Proceedings of the 24th

IDEAS 2020

Incheon/Seoul, South Korea

(On-line - Video conferencing)

August 12 – 14, 2020

IDEAS 2020

24th

International Database Applications & Engineering Symposium

Ɨncheon/Seoul, South Korea (On-line - Video conferencing)

2020-08-12 – 2020-08-14

Editor

Bipin C. Desai, Concordia University, Canada

General Chairs

Bipin C. Desai, Concordia University, Canada

Wan-Sup Cho, Chungbuk National University, Korea

Local Chair

Yoo-Sung Kim, Inha University, Korea

Kwan-Hee Yoo, Chungbuk National University, Korea

Program Chairs

Wookey Lee, Inha University, Korea

Carson Leung, University of Manitoba, Canada

Publicity Chairs

Aziz Nasridinov, Chungbuk National University

Younho Seong, NC A&T State University

Sponsors

Bytepress/ConfSys.org

Concordia University, Inha University

In co:operation with ACM

The Association for Computing Machinery

2 Penn Plaza, Suite 701
New York, New York 10121-0701

ISBN: 978-1-4503-7503-0

Editorial production by: BytePress

Cover Artwork by: (front) Letitia Desai (back) Bipin C. Desai

Table of Content

Full Paper

Full Paper(Continued)

Full Paper(Continued)

Full Paper(Continued)

Short Paper

Preface

These are the proceedings of the 24th annual event of IDEAS. We find the challenge of holding a quality conference is increasing with a larger number of meetings either underwritten by so called not-for-profit organizations with all invited paper or with papers guaranteed to be accepted. This meeting had the further challenge of dealing with the Corona virus pandemic. It is ironic that humankind had to endure some three major pandemics over the last century whereas IDEAS in its relatively short span of 25 years had to contend first with SARS and now this. SARS was contained and we were able to hold an in-person meeting in Hong Kong; however, this time we had to switch to a virtual meeting due to the virulent virus, It is heartening to learn that in spite of these challenges, we received 57 submissions. This allowed us to continue to be selective! This meeting highlights the current pre-occupation with AI, big data, block chain, data analytics, machine learning, the issue of a pandemic itself and issues with the web itself; this is reflected in the accepted papers in these proceedings.

I would like to take this opportunity to thank the general, program, local and publicity chairs and the program committee for their help in the review process. All the submitted papers were assigned to four reviewers and we got back over 2.6 reviews on the average due to the shorter review periods. The proceedings consist of 27 full papers(acceptance rate 47%), and 3 short papers (10%) .

Acknowledgment: This conference would not have been possible without the help and effort of many people and organizations. Thanks are owed to:

- ACM (Craig Rodkin, and Barbara Ryan),
- BytePress, ConfSys.org, Concordia University (Kunsheng Zhao, Gerry Laval, and Will Knight),
- Many other people and support staff, who contributed selflessly have been involved in organizing and holding this event.

I appreciate their efforts and dedication.

Bipin C. Desai
IDEAS/Concordia University
Montreal, Aug. 8, 2020

Reviewers
from the Program Committee

* Foto N Afrati(National Technical University of Athens, Greece)
* Ana Sousa Almeida(Instituto Superior de Engenharia do Porto, Portugal)
* Masayoshi Aritsugi(Kumamoto University, Japan)
* Ana Azevedo(Instituto Politecnico do Porto, Portugal)
* Gilbert Babin(HEC Montreal, Canada)
* Christopher Baker(University of New Brunswick, Saint John, Canada)
* Ayse Bener(Ryerson University, Canada)
* Giacomo Bergami(Newcastle University, United Kingdom)
* Jorge Bernardino(Instituto Politecnico de Coimbra, Portugal)
* Roi Blanco(Yahoo!, Spain)
* Christophe Bobineau(Institut National Polytechnique de Grenoble, France)
* Dumitru Dan Burdescu(University of Craiova, Romania)
* Gregory Butler(Concordia University, Canada)
* Ismael Caballero(Universidad de Castilla La Mancha, Spain)
* Richard Chbeir(Universite de Pau et des Pays de l'Adour, France)
* Rui Chen(Samsung, United States)
* David Chiu(University of Puget Sound, United States)
* Martine Collard(Universite des Antilles, France)
* Alfredo Cuzzocrea(University of Calabria, Italy)
* Gabriel David(Universidade do Porto, Portugal)
* Marcos Aurelio Domingues(Universidade Estadual de Maringa, Brazil)
* Brett Drury(Scicrop, Brazil)
* Magdalini Eirinaki(San Jose State University, United States)
* Markus Endres(Universitat Augsburg, Germany)
* Bettina Fazzinga(Consiglio Nazionale delle Ricerche, Italy)
* Alvaro Figueira(Universidade do Porto, Portugal)
* Sergio Flesca(University of Calabria, Italy)

Reviewers
from the Program Committee
(Continued)

* Alberto Freitas(Universidade do Porto, Portugal)
* Filippo Furfaro(University of Calabria, Italy)
* Pedro Furtado(Universidade de Coimbra, Portugal)
* Sven Groppe(Medizinische Universitat Lubeck, Germany)
* Antonella Guzzo(University of Calabria, Italy)
* Marc Gyssens(Hasselt University, Belgium)
* Irena Holubova (mlynkova)(Charles University Prague, Czech Republilc)
* Mirjana K Ivanovic(University of Novi Sad, Srebia and Montenegro)
* Nattiya Kanhabua(L3S Research Center, Germany)
* Will Knight(ConfSys.org, United States)
* Sotirios Kontogiannis(University of Ioannina, Greece)
* Michal Krátký(Technical University of Ostrava, Czech Republilc)
* Gerry Laval(ConfSys.org, Canada)
* Georgios Lepouras(University of Peloponnese, Greece)
* Carson K. Leung(University of Manitoba, Canada)
* Chuan-ming Liu(National Taipei University of Technology, Taiwan)
* Grigorios Loukides(King's College London, University of London, United Kingdom)
* Bertil P. Marques(Instituto Superior de Engenharia do Porto, Portugal)
* Elio Masciari(Consiglio Nazionale delle Ricerche, Italy)
* Mirjana Mazuran(POLITECNICO DI MILANO, Italy)
* Giuseppe M. Mazzeo(Facebook, United States)
* Richard Mcclatchey(University of the West of England, Bristol, United Kingdom)
* Hoda Mehrpouyan(Boise State University, United States)
* Peter Mikulecky(University of Hradec Králové, Czech Republilc)
* Dunja Mladenic(Jozef Stefan Institute, Slovenia)
* Noman Mohammed(University of Manitoba, Canada)

Reviewers
from the Program Committee
(Continued)

* Danilo Montesi(University of Bologna, Italy)
* Yang-sae Moon(Kangwon National University, Korea Republic)
* Kamran Munir(University of the West of England, Bristol, United Kingdom)
* Yiu-kai Dennis Ng(Brigham Young University, United States)
* Wilfred Ng(Hong Kong University of Science and Technology, Hong Kong)
* Mara Nikolaidou(Harokopio University, Greece)
* Selmin Nurcan(Universite Pantheon-Sorbonne (Paris I), France)
* Valéria Magalhães Pequeno(Universidade Autonoma de Lisboa Luis de Camoes, Portugal)
* Jaroslav Pokorny(Charles University Prague, Czech Republilc)
* Giuseppe Polese(University of Salerno, Italy)
* Luboš Popelínský(Masaryk University, Czech Republilc)
* Filipe Portela(Universidade do Minho, Portugal)
* Chiara Pulice(University of Calabria, Italy)
* Dimitrios Rafailidis(Maastricht University, Netherlands)
* Peter Z. Revesz(University of Nebraska - Lincoln, United States)
* Marina Ribaudo(University of Genoa, Italy)
* Filipe Rodrigues(Universidade de Coimbra, Portugal)
* Fereidoon Sadri(University of North Carolina at Greensboro, Reviewer)
* Maribel Yasmina Santos(Universidade do Minho, Portugal)
* Marinette Savonnet(Universite de Bourgogne, France)
* Younho Seong(North Carolina Agricultural and Technical State University, United States)
* Atsuhiro Takasu(National Institute of Informatics, Japan)
* Giorgio Terracina(University of Calabria, Italy)
* Stephanie Teufel(University of Fribourg, Switzerland)
* Motomichi Toyama(Keio University, Japan)
* Giuseppe Tradigo(University of Calabria, Italy)

Reviewers
from the Program Committee
(Continued)

* Goce Trajcevski(Iowa State University of Science and Technology, United States)
* Irina Trubitsyna(University of Calabria, Italy)
* Jeffrey David Ullman(Stanford University, United States)
* Domenico Ursino(Università Politecnica delle Marche, Italy)
* Costas Vassilakis(University of Peloponnese, Greece)
* Krishnamurthy Vidyasankar(Memorial University of Newfoundland, Canada)
* Eugenio Vocaturo(University of Calabria, Italy)
* Alicja Wieczorkowska(Polish-Japanese Institute of Information Technology in Warsaw, Poland)
* Carlo Zaniolo(University of California, Los Angeles, United States)
* Ester Zumpano(University of Calabria, Italy)

External Reviewers

IDEAS Steering Committee

Desai, Bipin C. (Chair)	*Concordia University*
McClatchey, Richard	*University of the West of England, Bristol*
Ng, Wilfred	*Hong Kong Univ. of Science and Technology*
Pokorny, Jaroslav	*Charles University*
Toyoma , Motomichi	*Keio University*
Ullman, Jeffrey	*Stanford University*

Patent Prior Art Search using Deep Learning Language Model

Dylan Myungchul Kang
Inha University
Incheon, South Korea
mckang1020@inha.edu

Charles Cheolgi Lee
Inha University
Incheon, South Korea
320056@inha.ac.kr

Suan Lee
VOICE AI Institute
Incheon, South Korea
suanlee@inha.ac.kr

Wookey Lee
Inha University & VOICE AI Institute
Incheon, South Korea
trinity@inha.ac.kr

ABSTRACT

A patent is one of the essential indicators of new technologies and business processes, which becomes the main driving force of the companies and even the national competitiveness as well, that has recently been submitted and exploited in a large scale of quantities of information sources. Since the number of patent processing personnel, however, can hardly keep up with the increasing number of patents, and thus may have been worried about from deteriorating the quality of examinations. In this regard, the advancement of deep learning for the language processing capabilities has been developed significantly so that the prior art search by the deep learning models also can be accomplished for the labor-intensive and expensive patent document search tasks. The prior art search requires differentiation tasks, usually with the sheer volume of relevant documents; thus, the recall is much more important than the precision, which is the primary difference from the conventional search engines. This paper addressed a method to effectively handle the patent documents using BERT, one of the major deep learning-based language models. We proved through experiments that our model had outperformed the conventional approaches and the combinations of the key components with the recall value of up to '94.29%' from the real patent dataset.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; **Natural language processing**.

KEYWORDS

Prior Art Search, Patent Document Classification, Language Model

ACM Reference Format:
Dylan Myungchul Kang, Charles Cheolgi Lee, Suan Lee, and Wookey Lee. 2020. Patent Prior Art Search using Deep Learning Language Model. In *Proceedings of 24th International Database Application & Enginnerring Symosium (IDEAS 2020).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.475/123_4

IDEAS 2020, August 12-18, 2020, Incheon, S. Korea(Virtual)

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

1 INTRODUCTION

As the number of patent applications for new technologies related to, so-called, the fourth industrial revolution, such as artificial intelligence (AI), Internet of Things (IoT), and big data, has rapidly grown, the scale of intellectual property rights, including patents in the global market, has tremendously been increased. In line with this global trend, several countries are attempting to increase patent prosecution quality while achieving patent rights fast and robust. For instance, switching from being a copycat, in February 2020, the Chinese telecommunications company 'Huawei' prosecuted the US company 'Verizon' for violating its patent rights on 12 technologies related to network security [5]. As such, the ownership of patents is becoming one of the most important measures of individual and business as well as the national competitiveness so that many of the companies have recently been encouraging and patenting the new technologies in huge quantities.

Compared to the increasing number of patent documents, the number of patent examiners and judges to handle them is not sufficient enough, and allocating the excessive workload to the limited workforce resources will lead inevitably to be deteriorated the quality of patent examination. Therefore, it is imperative for both the applicant and the examiner to perform the manual patent examination process both quicker and more accurate than before. In other words, the need for a smart patent search system is growing, and in order to satisfy these two goals; consequently, it is necessary to apply the rapidly developing deep learning language models.

The patent search tasks have the following several purposes. One of them is 'prior art search', that has been required before patent filing or for the prevention of patent infringement. It is significantly different for the patent search system on that the purpose and characteristics of the existing search engines have long been endowed. For example, the general search engines powered by Google or Yahoo have presented the search results from a user query, and if the user reviews the relevance of the results and finds that some of the relevant patents are missed, which is usually NO problem and the user seldom clicks up to the end of the dozens of millions of the results. Therefore, the precision measure or the precision of the precision is vital for general search engines, while the recall measure is much more critical for the patent search system. This is simply because of the possibility that if any patent-related document or technology item is missed in the prior art search, which may cause an unpredictably enormous amount of economic threat. Once we can see the huge money lawsuit case in the newspaper,

it is only when the patent settlement has been failed from among lots of settled cases. For this reason, the prior art search relies on a labor-intensive process in the order of grasping technology, extracting keywords, establishing search formulas, analyzing similar technologies, and writing formatted reports. Therefore, extremely high qualified expertise will strongly be required in deriving the valid patents, and at the same time, it will take fairly lots of time and effort. That is only one example of the differences between the two disciplines, and we reduce the whole story for the page limit. Therefore, we now acknowledge how the general search engines are different from the patent prior art search and the recall measure really matters in this search.

In this paper, we apply deep learning technology to patent prior art search. In particular, we deal with the binary classification problem of removing noise patents from the patent search system and finding valid patents by using a deep learning language model. This study proved that the deep learning language model showed a significant improvement in the prior art search. The followings are the brief structure of the paper. After discussing the related works in Section 2, the methodology with a deep learning language model is introduced in Section 3. The experimental steps are described in Section 4, and we conclude the paper in Section 5.

2 RELATED WORKS

2.1 Patent Document Classification

Recently, there have been studies to solve the labor-intensive document classification tasks with deep learning. Niu and Cai [8] attempted to classify CPC (Cooperative Patent Classification) codes, which is the extension of the IPC (International Classification Code) codes using a wide and deep neural network. In addition, Hu *et al.* [3] studied the problem of embedding the title and summary of patents using n-gram based CNN (Convolutional Neural Network), and classifying IPC codes through Bi-LSTM (Bidirectional Long Short-Term Memory). In the research [7], Li *et al.* classified IPC codes using CNN for patent documents embedded with word2vec. Lee and Hsiang [6] used the pre-trained deep learning language model BERT (Bidirectional Encoder Representations from Transformers) to classify CPC codes. Apart from that, some studies considered the characteristics of patent document search. Song *et al.* [11] achieved high recall through an effective method of finding relevant documents. In addition, Kang *et al.* [4] proposed a methodology for a user-friendly patent search system after precisely embedding a patent using bibliographic information and drawings constituting a patent document.

However, there is a critical problem with classifying CPC or IPC codes. In these papers, hierarchical features consisting of sections, class, subclass, main group, and subgroup that the classification code has not been taken into account. For instance, there is a patent that has 'G02B-021/08' as an IPC code. If the multi-label classification model outputs 'G02B-022/10', the evaluation measure regards it as a 'wrong' answer. The critical issue here is that even though the output is wrong, it is not desirable to count to a completely wrong answer because it is classified correctly up to 'G02B'. In other words, in order to deal with the problem of multi-label classification of patent code, a measure that well reflects the classification code characteristics with this hierarchical structure must be devised first.

Unlike the existing studies, therefore, this paper intends to address the binary classification problem of determining whether it is relevant in the prior art search process rather than multi-label classification problems such as IPC or CPC code classification. Secondly, the previous approaches have not considered the hierarchical feature of codes. IPC or CPC codes classification issues are the studies of how well the models fit the labels already set by patent examiners. However, searching valid patents with the binary classification issues in this paper is a much more important research topic since it can inform practitioners of what kind of the patents they should focus on in a practical manner, which can be expected to benefit the patent examination quality.

2.2 BERT Model for the Patent Document

In this section, we briefly describe BERT (Bidirectional Encoder Representations from Transformers) [2], which is excellent known that the model still performs well in natural language processing and because the patent documents per se are written by languages. BERT has been pointed out that the unidirectional learning model (OpenAI GPT, Generative Pre-Training [10]) and the shallow bidirectional learning model (ELMo, Embeddings from Language Models [9]) have revealed the language representation perspective not fully enough. Therefore, BERT pretrained large-scale corpus in a deep and bidirectional method and fine-tuning without adding a new network and achieved state-of-the-art results in many NLP tasks.

The BERT model's inputs consist of three embeddings: token embeddings, segment embeddings, and position embeddings. First, all words in the input sentence are tokenized through WordPiece embeddings [13]. For the beginning of every sentence, '[CLS]' token is added, and the hidden state value of the last layer of this token is used as a representation of the entire sentence. For the other tasks, the token is not used. In addition, the two sentences are divided into '[SEP]' token, and different segment embeddings are given to the front and back sentences. Lastly, unlike the RNN (Recurrent Neural Network) model, the BERT takes sentences in parallel, so the model cannot contain relative position information. Positional embedding is used to tackle this problem. Subsequently, all three embeddings representing one token are added and used as input for the model.

Now, we will explain the training process of the BERT model. BERT fine-tunes specific tasks after pretraining on a large corpus. The datasets used to pretrain the model are 800M words from BooksCorpus and 2,500M words from English Wikipedia. The input value in the training process refers to 'sentences' and is a pair of two sentences in which the sum of the sentences' lengths is less than 512. The BERT model is trained by 'Masked LM', a method of randomly selecting 15% tokens, and 'Next Sentence Prediction', matching whether a sentence is likely to appear in the next sentence in order to learn the relationship between sentences. Specifically, in 'Masked LM', 80% of the selected tokens are converted to '[MASK]' tokens, 10% to other random words, and 10% remain. Also, in the Next Sentence Prediction, 50% show the actual next sentence, and 50% do not.

In this regard, the patent prior art search has been treated and adopted in terms of the pretrained language model BERT for the first time in this paper.

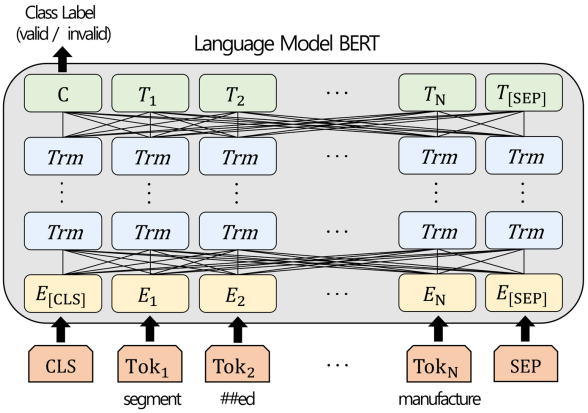


Figure 1: The architecture of the BERT model with fine-tuning for the classification task

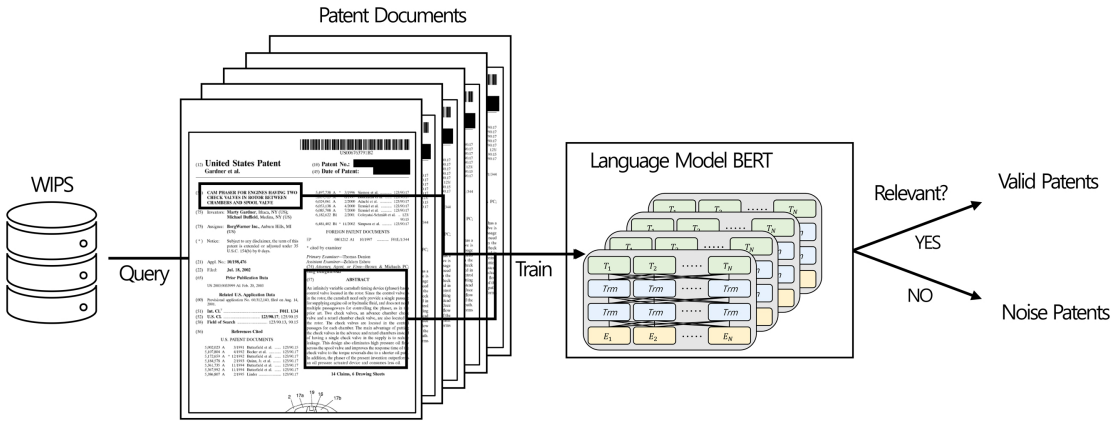


Figure 2: The framework of patent prior art search using BERT model

3 METHODOLOGY

Now, we would like to explain the attention mechanism used in the framework of this paper. The basic elements are described in [12]. The deep learning language model BERT processes tokens containing textual information as the patent's components p using the encoder part of the transformer. BERT used "Scaled Dot-Product Attention" that can be described as follows.

$$Attention(Q_p, K_p, V_p) = softmax(\frac{Q_p K_p^T}{\sqrt{d}})V_p \quad (p \in P) \quad (1)$$

First, we generate queries, keys, and values for the p respectively, and compute the dot products of all queries and keys and divide them in $\sqrt{d}$ where d represents the size of queries and keys. After that, we apply the softmax function to it and multiply it by the values. In practice, the queries, the keys, and the values are packaged together into matrices Q_p, K_p, and V_p respectively as explained in Equation 1.

The patent bibliographic information set P includes the components such as title, abstract, and claim, and the combination of these. Therefore, k of components p belongs to set P. That is, $p \in P$, and the size of the set P can be defined as follows. Here, C indicates 'combination' which is a selection of items from a collection.

$$|P| \leq \sum_{i=1}^{k} {}_kC_i \quad (2)$$

Now, in Figure 1, we will explain the process of taking the title of a patent document into the BERT model. A preprocessing process is performed to use textual information of patent documents in a deep learning model, which includes removing special characters and numbers. For example, here is the patent title, ['Segmented multi-focal contact lens and method of manufacture'], and the tokenized patent title is demonstrated in Figure 1, ['segment', '##ed', 'multi', '##fo', '##cal', 'contact', 'lens', 'and', 'method', 'of', 'manufacture']. Additionally, for each token, we assigned IDs, and to use the BERT model as a classification model, the token '[CLS]' is added at the beginning of each sentence, and the token '[SEP]' is added at the end.

As shown in Figure 2, the binary classification model for patent prior art search performs a classification task using a deep learning language model using only bibliographic information of patent

Table 1: A result table of the experiment

Dataset	Section	Measure			Validation Loss			
		Precision(%)	Recall(%)	F1(%)	epoch1	epoch2	epoch3	epoch4
Non-uniform	title	62.50	49.02	54.95	0.14	0.14	0.16	0.17
	abstract	56.76	52.50	54.55	0.12	0.10	0.11	0.12
	first claim	63.16	40.00	48.98	0.14	0.11	0.08	0.06
	title + abstract	46.34	57.58	51.35	0.17	0.12	0.14	0.15
	title + first claim	63.33	45.24	52.78	0.09	0.09	0.08	0.09
Uniform	title	60.47	78.79	68.42	0.52	0.42	0.37	0.34
	abstract	74.72	88.89	81.01	0.56	0.40	0.35	0.34
	first claim	**81.58**	86.11	**83.78**	0.71	0.62	0.56	0.56
	title + abstract	71.43	85.71	77.92	0.52	0.37	0.30	0.30
	title + first claim	71.74	**94.29**	81.48	0.58	0.44	0.41	0.43

data. To this end, the patent documents derived from the online patent information retrieval system 'WIPS [1]' with queries for dual-camera technology were used as a dataset. We added labels 0 (noise patent) and 1 (valid patent) to the derived patent documents to meet the purpose of the prior art search. The preprocessed patent documents are trained in a binary classification BERT model, and then the performance is evaluated through test data.

4 EXPERIMENTAL RESULT

4.1 Environment

The experiment was conducted in Google's 'Colaboratory' environment, and GPU used 'Tesla K80', 'Tesla P4', and 'Tesla P100-PCIE-16GB' and 'Intel(R) Xeon(R) CPU @ 2.20GHz / 2.30GHz' depending on the server situation. We used four epochs and 32 batches for all experiments, and the maximum length entered into the BERT model was 64 for the title data and 128 for the rest. In addition, we used 'AdamW' as an optimizer with the epsilon 1e-8, and the learning rate was 2e-5.

4.2 Dataset and Search Query

Table 2: Dataset used in the Experiment

Dataset	Training		Test		Total
	Noise Patent	Valid Patent	Noise Patent	Valid Patent	
Non-uniform	7,832	351	859	51	9,093
Uniform	364	359	38	43	804

The dataset used in the experiment was fetched from 'WIPS' by March 2016 through the search query as followings which originated from the Equation 3, including 'US (United States)', 'CN (China)', 'EP (Europe)', and 'PCT (Patent Cooperation Treaty)' patents. Each patent has the properties of nation code, title, abstract, first claim, application number, main IPC code. We add label 1 to the patent judged to be relevant and added label 0 to the patent judged not to be relevant.

$$((plural* \ OR \ dual* \ ... \ transfer* \ OR \ barrel* \ OR \ signal*) \quad (3)$$

The data used in the experiment were 'Training:Validation:Test' and the ratios were divided into '8:1:1' respectively, and the total number of data may vary because of missing values for summaries and claims, but the ratio was the same. The different number of data was utilized for two experiments, uniform and non-uniform. Up to 9,093 for non-uniform datasets and up to 804 for uniform datasets were used depending on missing values. Because valid patents (label 1) are only 4.4% of the total data, the ratio of valid patents and noise patents was set to 1:1 in the uniform dataset to evaluate the model's performance.

4.3 Result

A total of 10 experiments were conducted, and the results are shown in Table 1 using the uniform and non-uniform dataset. We used 'bert-base-uncased' language model that has 12 layers of transformer blocks. As previously explained, the recall is important in the prior art search. Therefore, the results should be viewed on a basis with a high recall rate, and the use of the title and the first claim gave the best performance at 94.29% on the uniform dataset. Besides, it can be said that the first claim is a relatively good representation of the contents of the patent, given that the experiment using only the first claim has an 86.11% recall rate. Even from a practical point of view, claims play a significant role in patentability, which has been demonstrated through the experiment. One of the features of patent documents is the use of ambiguous expressions to broaden the scope of the claim. These features are common in claims, and it is essential to review claims for a precise understanding of patents. Besides, claims have a unique structure, which often appears in legal documents. Claims have a hierarchical structure because they are written while adding an explanation to the previous claim term. As the claims are indispensable in the patent, it should be studied using the structural properties of the claims.

On the right side of the 1 is the validation loss. In some cases, the loss value does not decrease during the training of the non-uniform dataset. This means that the language model is not appropriate for learning uneven data. (It may simply be due to a small number of data.) On the contrary, steadily decreasing or maintaining the loss

value for the uniform dataset indicates that the model has been well trained without overfitting. Accordingly, if the recall-oriented classification model can be appropriately designed for non-uniform real patent data, it is expected to strengthen patent examination quality and accelerate the examination process from a practical point of view during prior art search.

5 CONCLUSION

As the importance of the role of the patent being highlighted, the prior art search and analyses of the patent documents have also tremendously been captured the researchers' attention. In this paper, we tackled the problem of finding a valid patent during the prior art search with the deep learning language model. We can effectively distinguish the relevant patents from eliminating the invalid patents by actively adopting the deep learning language model BERT among the patent datasets. Besides, the experiment achieved a high recall rate of '94.29%' for uniform data, which resulted in the patent practitioners reducing the examination period and meeting the original purpose of finding related and valid patents. Therefore, we have shown that the approaches of this paper could contribute to the selection of significant patents and search for prior technologies with nearly 5% errors which can help the inventors, patent attorneys, examiners and be useful not only to the prior art search, but also to patentability, patent invalidation, patent transactions, compensations, and infringement lawsuit efforts.

In the future, the language model architecture should be adapted to the patent document. We plan to use metadata such as citations and build multi-modal patent embedding to carry out a more effective patent search.

6 ACKNOWLEDGMENTS

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2019-0-00136, Development of AI-Convergence Technologies for Smart City Industry Productivity Innovation)

REFERENCES

[1] *WIPS ON*, accessed May 1st, 2020. http://www.wipson.com/service/mai/main.wips.
[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
[3] J. Hu, S. Li, J. Hu, and G. Yang. A hierarchical feature extraction model for multi-label mechanical patent classification. *Sustainability*, 10(1):219, 2018.
[4] M. Kang, S. Lee, and W. Lee. Prior art search using multi-modal embedding of patent documents. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 548–550. IEEE, 2020.
[5] A. Kharpal. Chinese telecoms giant huawei sues verizon for patent infringement. *CNBC*, Feb 2020 (accessed May 1st, 2020).
[6] J.-S. Lee and J. Hsiang. Patentbert: Patent classification with fine-tuning a pre-trained bert model. *arXiv preprint arXiv:1906.02124*, 2019.
[7] S. Li, J. Hu, Y. Cui, and J. Hu. Deeppatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117(2):721–744, 2018.
[8] M. Niu and J. Cai. A label informative wide & deep classifier for patents and papers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3429–3434, 2019.
[9] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
[10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.
[11] J. J. Song, W. Lee, and J. Afshar. An effective high recall retrieval method. *Data & Knowledge Engineering*, 123:101603, 2019.
[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
[13] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Bridging the Gap Between Business Processes and IoT

Zakaria Maamar
Zayed University
Dubai, UAE

Ejub Kajan
State University of Novi Pazar
Novi Pazar, Serbia

Ikbel Guidara
Claude Bernard Lyon 1 University
Lyon, France

Leyla Moctar-M'Baba
SAMOVAR, Télécom SudParis
Institut Polytechnique de Paris
Evry, France

Mohamed Sellami
SAMOVAR, Télécom SudParis
Institut Polytechnique de Paris
Evry, France

ABSTRACT

This paper discusses a novel way of making business processes and Internet of Things (IoT) work together. Indeed each "suffers" from many limitations that the other could help address them and *vice versa*. On the one hand, business processes are known for capturing organizations' best practices when satisfying users' demands but do not have the capabilities of controlling the physical surrounding that comprises millions of devices/things. On the other hand, IoT is known for provisioning contextualized services to users thanks to millions of devices/things but does not have the capabilities of making these devices/things collaborate. The paper presents a framework to support the collaboration of business processes and IoT. This collaboration is exemplified with 2 types of processes referred as thing-aware processes (TaP) and process-of-things (PoT). A system illustrating the development of PoT is presented in the paper as well.

CCS CONCEPTS

• **General and reference** → General conference proceedings; • **Applied computing** → **Business process modeling**; • **Software and its engineering**; • **Computer systems organization** → *Embedded and cyber-physical systems*;

KEYWORDS

Business Process, Collaboration, Context, Internet of Things.

ACM Reference Format:
Zakaria Maamar, Ejub Kajan, Ikbel Guidara, Leyla Moctar-M'Baba, and Mohamed Sellami. 2020. Bridging the Gap Between Business Processes and IoT. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410605

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06.
https://doi.org/10.1145/3410566.3410605

1 INTRODUCTION

Commonly referred to as *know-how*, a Business Process (BP) *"... consists of a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal. Each business process is enacted by a single organization, but it may interact with business processes performed by other organizations."* [34]. A simple definition is that a BP consists of activities connected together with respect to a process model that specifies who does what, where, and when. Examples of BPs include product procurement and loan approval.

To ensure a competitiveness advantage, organizations constantly reengineer their BPs because of users' changing needs and requirements and advances in Information and Communication Technologies (ICT) such as blockchain, cloud/edge, 5G networks, and Internet-of-Things (IoT). Indeed some ICT like blockchain improves security while others like cloud reduces upfront capital expenditures. However, to sustain this competitiveness advantage, proper integration of ICT into BP should take place so that disappointments are mitigated, failures are avoided, and efforts pay off. In this paper we adopt IoT to exemplify ICT and present how IoT and BP collaboration would conceptually and technically take shape.

According to Gartner[1], 6.4 billion connected things were in use in 2016, up 3% from 2015, and will reach 20.8 billion by 2020. The wireless world research forum also reports that in 2017, there were 7 trillion wireless devices forming a complete IoT ecosystem [24]. This large and ever-growing number of things need to be "harnessed" so that things' collective over individual behaviors prevail [15]. IoT is a perfect demonstration of Weiser's definition of ubiquitous computing when he states in 1999 that "*the most profound technologies are those that **disappear**. They weave themselves into the fabric of everyday life until they are indistinguishable from it*" [33].

We foresee BP-IoT collaboration as a win-win situation for both. We examine how IoT could help address some BPs' limitations and *vice-versa*. On the one hand, BPs could tap into IoT to have a better control over the physical surrounding in which these BPs run. On the other hand, IoT could tap into BPs to have a better coordination over isolated things

[1] www.gartner.com/newsroom/id/3165317.

(confined into "silos"). Initiatives that support BP-IoT collaboration include the 2^{nd} International Workshop on BP-Meet-IoT[2] and the works of Brouns et al. [3], Maamar et al. [16], and Seiger et al. [28]. Soto considers the merge of IoT and BP as a signal of true disruption[3].

Our contributions are, but not limited to, (i) definition of a collaborative framework for BPs and IoT, (ii) discussion of how IoT addresses BPs' limitations and *vice-versa*, (iii) illustration of how BPs and IoT act jointly, and (iv) partial implementation of the collaborative framework. The rest of this paper is organized as follows. Section 2 motivates BP-IoT collaboration. Section 3 summarizes some related works. Section 4 presents the framework that achieves this collaboration. Section 5 discusses this framework implementation. Section 6 concludes the paper.

2 MOTIVATIONS

To motivate BP-IoT collaboration we consider a city that runs many systems (sys) like Transportation ($\mathcal{T}_{sys}$) for traffic control and Environment ($\mathcal{E}_{sys}$) for air pollution monitoring. The city has to deal with a temporarily closed tunnel following a car accident (Fig. 1). Initially, the tunnel's cameras report the accident to $\mathcal{T}_{syst}$ that proceeds with informing drivers using variable-message signs. One of the accident's consequences is that the carbon-dioxide level inside the tunnel rises up making $\mathcal{E}_{sys}$ issue warnings to drivers and emergency services.

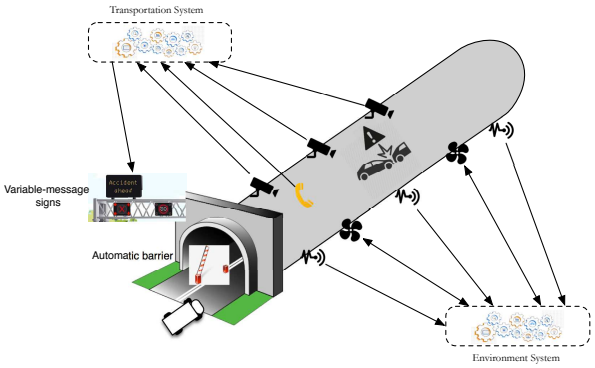


Figure 1: BP-IoT collaboration during tunnel closure

This simple, yet, realistic case-study illustrates how BPs and IoT can collaborate together. On the one hand, the BPs to manage the tunnel closure would indicate the activities to perform like deviating traffic, contacting emergency services, informing drivers, etc. Some activities could easily connect to (physical) things. For instance, deviating traffic makes road signs display a new speed limit and contacting emergency services makes variable-message signs post relevant messages. On the other hand, the IoT ecosystem of the tunnel would indicate the necessary (physical) things to involve like cameras, fans, signs, etc. Some things could easily connect to BPs' activities. For instance, sensors increase the streaming rate of air quality so that more data is sent to $\mathcal{E}_{sys}$'s activities for processing and fans increase the rotation speeds so that $\mathcal{T}_{sys}$'s activities opens the tunnel's barrier.

BP-IoT collaboration (through activity-thing collaboration) cannot happen without first, identifying the activities that should be "alert to" what things are doing and second, identifying the things that should be "responsive to" activities' outcomes. We recall that

- Activities of BPs are defined with respect to a controlled environment in the sense that who does what, when, and where are pretty much known. Could things help activities react to unexpected changes?
- Things in IoT operate in an open environment where changes constantly happen. Could activities help things decide about what to do with these changes and how to handle these changes?

3 BRIDGING THE GAP

This section discusses some related works and then, discusses how BPs' and IoT's concerns overlap.

3.1 Overview of some works

At the time of this writing our literature review resulted into a limited number of works that touch upon BP-IoT collaboration and are in line with the way we envision this collaboration. 3 particular works drew our attention:

- In [12], Hasić and Serral Asensio shed light on the current support and existing challenges of executing IoT processes modeled in Business Process Model and Notation 2.0 (BPMN). Firstly, the current support is associated with some BPMN 2.0 constructs that process engineers would use to model IoT scenarios like service task, business rule task, event, resource, and data. Secondly, the existing challenges are about IoT resource awareness of tasks, IoT resource binding at run-time, IoT resource malfunction event-handling, IoT data retrieval (pull *versus* push), and IoT tasks and human resources communication.
- In [17], Mandal et al. develop a classification of BP-driven IoT scenarios and identify some benefits of injecting Business Process Management (BPM) concepts into these scenarios. The classification is based on 6 categories of parameters that are participant (specialized into sensor, actuator, display, controller, complex device, Web service, and human being), control (specialized into central, on-device, and distributed), interaction (specialized into thing-to-thing and thing-to-controller), data, automation, and ownership.
- In [22], Mukherjee et al. discuss IoT workflow development. This workflow progresses with respect to human triggered state changes as well as triggers based on IoT messages. The authors also discuss how today's

[2] www.pros.upv.es/sites/bp-meet-iot2018.
[3] tinyurl.com/yyx7osgt.

things are equipped with sensors and hence, the messages they send permit to monitor them and track their state changes. Mukherjee et al. address the cyber-physical separation using sensor data.

It is clear that none of the 3 works discuss how BPs could address some IoT's limitations nor how IoT could address some BPs' limitations. Despite the potential benefits of BP-IoT collaboration, both are still loosely-connected [11] having each separate concerns and priorities that, in fact, overlap at a certain stage of their respective lifecycles. On the one hand, BPs capture organizations' best practices that need to be enforced and aligned with any regulation. However, because of humans' potential participation in BPs, humans do not always comply with BPs leading sometimes to unacceptable deviations. Would IoT help address these deviations, for example? On the other hand, IoT is backed with the continuous development of communication technologies allowing more data about things to be transferred in a timely and secure manner. However, because of things' limitations like reduced size, restricted connectivity, continuous mobility, limited energy, and constrained storage, things have been confined into a role of data suppliers. Would BPs help change this role, for example? In this paper we address these 2 questions on top of the other 2 raised at the end of Section 2. We discuss the support of BPs to IoT in Section 3.2 and the support of IoT to BPs in Section 3.3.

3.2 When BPs meet IoT

Different limitations undermine the benefits of adopting IoT by the ICT community. These limitations are mainly related to the diversity and multiplicity of things' development and communication technologies [1] and limited IoT-platform interoperability [2]. Below we discuss how BPs could address 3 particular IoT-related limitations that we refer to as *silo*, *no-reasoning*, and *no-proper modeling technique*.

Silo restriction. Due to their passive nature [8], things "barely" collaborate which is not in line with the nature of today's applications. To allow things to collaborate, BPs could "cement" the collaboration by defining data dependencies and message exchanges between things and specifying roles of things. Below are some works that address the silo restriction.

- In [16], Maamar et al. introduce the concept of Process-of-Things (PoT) to enable the support of BPM to IoT. By analogy to a BP that has a process model, a PoT has a story [32]. It indicates the necessary things that would collectively act-upon the cyber-physical surroundings of users. A PoT specializes things into living and non-living, assigns purposes to non-living things and roles to living things, makes things rehearse prior to their activation, and evaluates the performance of things after their activation. In PoTs, stories refer to scripts that things play, characters that things embody, and scenes (*aka* locations in [11]) where things act.
- In [27], Seiger et al. suggest HoloFlows so that things participate in common processes. The authors resort to augmented reality to develop an app for smart glasses. Users of these glasses explore the surroundings and model some basic workflows that would connect sensors and actuators together using virtual wires.
- In [30], Suri et al. discuss the concept of IoT-aware process by developing a framework that describes IoT resources, formalizes IoT properties and rules for optimal resource management in processes, and addresses resource-based conflicts. The authors propose a semantic model called Internet of Things in Business Processes Ontology (IoT-BPO). IoT-BPO re-uses and extends some IoT-Lite concepts. The framework also helps ensure error-free allocation of heterogeneous IoT resources to BPs to foster interoperability.

No-reasoning restriction. According to a 2015 IBM white-paper [10], IoT needs to be smarter so that better results from things are achieved. This smartness could happen thanks to cognitive computing. In a similar statement, Wu et al. argue that "*without comprehensive cognitive capability, IoT is just like an awkward stegosaurus: all brawn and no brains*" [35]. Brain-empowered IoT or Cognitive Internet-of-Things (CIoT) are the terms that Wu et al. use to describe the future generation of things. Below are some works that address the no-reasoning restriction.

- In [13], Maamar et al. blend cognitive computing with IoT to develop cognitive things. They argue that today's things are confined into a data supplier role, which deprives them from being the technology of choice when developing advanced smart applications, for example. Cognitive computing would allow things to reason, adapt, and learn.
- In [31], Vlacheas et al. mention that a cognitive management of things would need answers to many questions like how things should be connected, why and when things need to be connected, and what value things can bring to existing services and applications. The authors state that "*cognitive technologies are about the ability to dynamically select behaviors through self-management, taking into account information and knowledge on the context of operation as well as policies*".

No-proper modeling technique restriction. Despite the popularity of IoT, there is not an IoT-oriented software engineering discipline that would guide the analysis, design, and development of IoT applications [6, 36]. BPs' existing modeling techniques like BPMN could be either adopted as-is or adapted to model such applications. Below are some works that address the no-proper modeling technique restriction.

- In [11], Grefen et al. discuss the specification of time and space when modeling IoT-aware collaborative BPs. The authors refer to both time and space as co-location that identifies physical objects to synchronize so that digital processes are successfully achieved. Grefen et al. distinguish 2 kinds of geographic co-location, absolute and relative, and use the concept of sphere to model this co-location. They adopt the same for time co-location in terms of absolute and relative types and

sphere use. Consistency between geographic and time spheres is considered in this work as well.
- In [30], Suri et al. extend BPMN to connect IoT resources to BPs. The objective is to capture both the structured nature of BPs and the structured nature of IoT devices and their specific characteristics/properties, and to build relationships between BP concepts, mainly tasks and their resource allocation.
- In [20], Meroni discusses the integration of IoT into BPM and notes that this integration runs into many problems related to process compliance and smart object configuration. From an IoT perspective, smart objects are devices that permit to decentralize data computation and acquisition. A smart object is equipped with a sensor network, a single board computing unit, and a communication interface. Meroni enriches BPMN with constructs that depict smart objects' roles and needs inside BPs.
- In [21], Meyer et al. integrate IoT devices into BPs as resources. IoT devices like sensors interact with the physical surrounding and hence, could feed processes with relevant, live data. However, Meyer et al. note the limited handling of IoT devices' characteristics during process modeling. To address this limitation, they suggest mapping the main abstractions and concepts of the IoT domain (namely IoT service, physical entity, IoT device, and native service) onto specific notations and constructs. The concept of IoT devices as resources is also discussed by Cheng et al. who propose extending BPMN meta-model to support the direct modeling of these devices [4].
- In [7, 19], Domingos and Martins use BPMN to model the behaviors of IoT devices. Relevant BPMN elements include control flow (that includes events, activities, and gateways), connecting objects (that include sequence flow, message flow, and data association), and data (that includes data objects). Once an IoT device's behavior is set, it is translated into Callas bytecode that is a sensor programming language.

3.3 When IoT meets BPs

By analogy to Section 3.2 we discuss how IoT could address 3 particular BP-related limitations that we refer to as *physical surrounding*, *data input*, and *context insensitivity*. According to Seiger et al. [28], making existing workflow management systems autonomous is important so they can handle unanticipated situations that could emerge in the context of IoT.

Physical-surrounding restriction. BPs operate in a cyber surrounding and thus, have limited (or even no) access to the physical surrounding. Contrarily, IoT evolves in a cyber-physical surrounding so BPs could access the physical surrounding through IoT. According to Meyer et al. [21], the physical surrounding is necessary for process models to ensure the correct resolution and execution of processes. Below are some works that address the physical-surrounding restriction.

- In [9], Friedow et al. integrate and coordinate IoT devices into BPs so that events that IoT devices generate can influence the execution of process instances and process instances can send commands to IoT devices and change the state of the physical surrounding. The authors use a process engine to define a process layer for IoT applications.
- In [21], Meyer et al. extend the BP modeling standard, BPMN 2.0, to capture IoT devices and identify their software components as process resources to the BP model. The objective is to include physical world resources in BPs so that these BPs execution is enhanced. The authors propose a semantic model to capture resource allocation requirements that will be used in later BPM phases. Then, they propose a mapping of the main relevant IoT components (IoT service, IoT device, physical entity, and native service) onto those provided by BP modeling notations.

Data-input restriction. BPs' activities are expected to act upon data that IoT could provide. In fact IoT-related data could be used to take actions (e.g., trigger alerts) that would allow BP execution to progress [5]. Below are some works that address the data input restriction.

- In [5], Cherrier et al. propose a software gateway that would allow BPs to access IoT data. The gateway transfers IoT events as an input to a BP while managing the heterogeneity of IoT devices and facilitating access data in terms of response time, throughput, and failure. Data gathered and computed by the gateway are changed into events that are transmitted to the BP. The objective is to establish a bridge that would make IoT devices communicate with the BP while hiding their specificities.
- In [25, 26], Schönig et al. propose an approach that exploits IoT data for BPM and use these data to interact with BPs. The approach extends a BPMN process model with IoT data variables. First, data are modeled as an ordered sequence of events. Then, a semantic mapping between IoT object data and process variables is defined. Finally, an extension of the BPMN diagram is proposed to include details about connected IoT variables (e.g., IoT-based trigger events, IoT-based decisions using data-based gateway, and IoT-based loops to model repeated behavioral patterns). The approach shows how the integration of IoT data into BP enhances a BPMS.

Context-insensitivity restriction. BPs are insensitive to context. Contrarily IoT is by default context sensitive through sensors and thus, could make BPs "appreciate" the physical surrounding. BPs need to know the conditions and situations

in which IoT operates. Context may include details about devices, sensors, places, users, etc. Below are some works that address the context-insensitivity restriction.

- In [26], Schönig et al. discuss how they distribute IoT data over BP users in a context-aware manner and how they provide user interfaces with context specific IoT data to improve the quality of activity execution. The authors suggest 3 dimensions when distributing IoT data over BP participants: dedicated context to select the appropriate entities for each specific context, information/source to define which IoT data should be provisioned, and visualization to indicate how context-specific IoT data must be presented.

4 BP-IOT FRAMEWORK

This section consists of 3 parts. The first part discusses the BP-IoT collaboration framework's components and the last two parts illustrate how these components action the framework.

4.1 Components of the framework

Fig. 2 illustrates our framework for BP-IoT collaboration. The framework spreads over 2 worlds, cyber and physical, and consists of 2 repositories, 6 modules, and 2 platforms. Each resides in a particular world but they all contribute to the operation of what we here refer to as either BP Management System (BPMS)[4] or IoT Management System (IoTMS).

BP management system. In the cyber world, BP engineers use the repository of activities as an input to the design module in order to model[5] the process models of future BP applications (e.g., toll payment in the tunnel and maintenance scheduling of the tunnel). We recall that a BP consists of activities and dependencies (e.g., finish-to-start and start-to-start) between activities. After completing the BP design, the BP engineer through the exchange module (and in collaboration with the configuration module of the IoTMS) identifies the necessary things that would allow the newly-designed BP to act upon the physical world and hence, to address (some) BPs' limitations discussed in Section 3.3. In fact, the engineer makes activities (those deemed necessary) connect to specific things which we demonstrate in Section 5. This connection leads to the formation of a Thing-aware Process (TaP) and is taken care by the binding module that is located in the physical world. Finally, the TaP is deployed on a BP execution platform located in the physical world.

Fig. 3 illustrates a simple TaP where activities like a_1 and a_i bind to things such as t_1 and t_2/t_k, respectively. Binding allows a TaP's activities (i) to execute commands over things like setting sensing frequency in the tunnel and (ii) to collect data from things. In Fig. 3, things are contextualized (C) allowing activities to have a better appreciation of the physical world in which the things reside.

[4]It enhances existing BPMS's traditional functionalities.
[5]We recommend the standard BPMN for BP modeling.

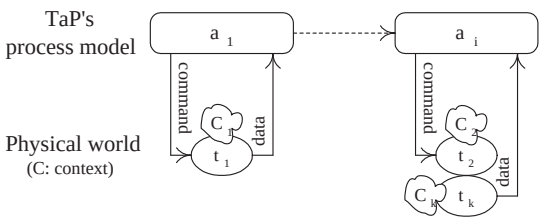


Figure 3: Illustration of a thing-aware process (TaP)

IoT management system. In the cyber world, IoT engineers use the repository of things as an input to the configuration module in order to get things ready for operation (e.g., setting-up the tunnel's fans). We recall that at this stage things are still independent from each other. After configuration, the IoT engineer through the exchange module identifies the necessary BP (in fact, its process model) that will allow to put the newly-configured things together and hence, to address (some) IoT's limitations discussed in Section 3.2. To this end, the engineer injects the BP's process model into things which leads to the formation of a Process of Things (PoT). This injection is taken care by the weaving module that is located in the physical world. Finally, the PoT is deployed on an IoT execution platform located in the physical world as well.

Fig. 4 illustrates a simple PoT where things like t_1 and t_i (associated with a_1 and a_i, respectively) are connected together according to a specific process model that has a set of activities $(a_1, a_2, ..., a_i)$ and is obtained thanks to the BPMS's exchange module. This process model defines the exchange of messages between things and the commands to initiate over things. In Fig. 4, things are contextualized (C) allowing them to reason about themselves and the messages they receive from peers.

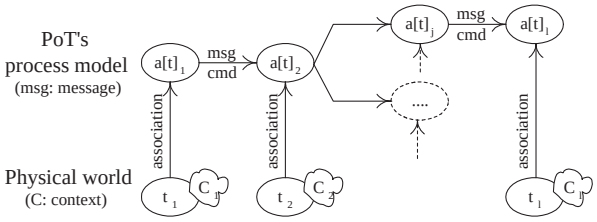


Figure 4: Illustration of a process of things (PoT)

4.2 BPMS's and IoTMS's functionalities

BPMS and IoTMS are the pillars upon which the BP-IoT collaboration framework is built. Whilst BPMS is well defined in the literature, IoTMS remains undefined but could benefit from Sheng et al.'s statement that *"management generally consists of configuration, monitoring and administration of managed entities"* [29]. Briefly, a BPMS is responsible to *"coordinate an automated business process in such a way that all*

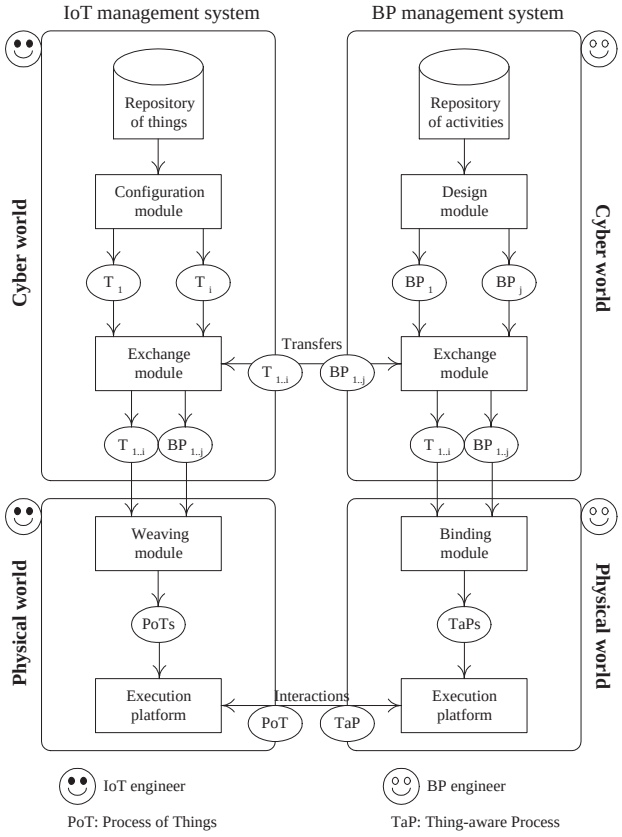


Figure 2: Framework for BP-IoT collaboration

work is done at the right time by the right resource" [18]. By analogy to BPMS, we would make IoTMS coordinate things with respect to triggered events, exchanged messages, and initiated commands.

In preparation for actioning BPMS and IoTMS, we recommend functionalities for each while considering the respective limitations of BPs and IoT. We proceed with the BPMS's functionalities that would be achieved in collaboration with the IoTMS.

(1) Data functionality: supporting TaPs, the BPMS should assist with receiving (real-time) data from things so that TaPs process these data. Examples of data include rotation speeds of fans, air quality in the tunnel, and ambient temperatures. Data could also be used to initiate TaPs, trigger TaPs' activities, and select decisional branches in TaPs. The BPMS's data functionality addresses the data-input restriction.
(2) Context functionality: supporting TaPs, the BPMS should assist with collecting contextual details about things and their cyber-physical surroundings. Details could be about cameras' and fans' locations to support the execution of TaPs such as opening the tunnel's barrier. The BPMS's context functionality addresses the context-insensitivity restriction.
(3) Accessibility functionality: supporting TaPs, the BPMS should assist with analyzing things' technical configurations (e.g., sensor's type and communication protocol) so that TaPs execute commands over these things and collect data from these things. The BPMS's accessibility functionality addresses the physical-surrounding restriction.

We now define the IoTMS's functionalities that would be achieved in collaboration with the BPMS.

(1) Connection functionality: supporting PoTs, the IoTMS should assist with "gluing" things together using process models that would connect things and activities together, define dependencies between things, and identify triggers of things. The IoTMS's connection functionality addresses the silo restriction.
(2) Cognitive functionality: supporting PoTs, the IoTMS should monitor all under-execution PoTs so that things can reason about what was performed, with whom they interacted, and what was shared/communicated.

The IoTMS's cognitive functionality addresses the no-reasoning restriction.

(3) Accessibility functionality: supporting PoTs, the IoTMS should assist with analyzing process models so that commands to exchange between things are identified and messages to execute between things are identified too. The IoTMS's accessibility functionality addresses the no-proper-modeling restriction.

4.3 When BPMS-IoTMS act jointly

Fig. 2 includes 2 links that bridge the gap between BPs and IoT thanks to BPMS's and IoTMS's respective functionalities and modules that reside in either the cyber world or the physical world.

- The 1^{st} link, *transfer*, is specialized into *T-transfer* that "ships" necessary things to under-development TaPs, and *BP-transfer* that "ships" necessary process models to under-development PoTs.
- The 2^{nd} link, *interaction*, is specialized into *BP-interaction* that allows TaPs to convert (relevant parts of) process models into commands for invoking things that will run on top of the IoT execution platform and hence, act upon the physical surrounding, and *T-interaction* that allows PoTs to identify things' commands to execute on top of the BP execution platform and hence, act upon the cyber surrounding.

In the following we discuss how *transfer* and *interaction* links act jointly to form TaPs and PoTs (Algorithms 1 and 2). We first begin with *transfer*.

- *T-transfer* link "ships" the descriptions of things to TaPs by involving the IoTMS's exchange module as a sender and the BPMS's exchange module as a recipient. Upon description receipt, the BPMS's context functionality processes this description so that details about things like current operations and ongoing participation in other TaPs are extracted. The objective of analyzing these details is to establish (activity,thing) couples like shown in Fig. 3.
  In the BP-IoT framework, a thing description refers to functional and non-functional properties that are based on our Quality-of-Thing (QoT) model discussed in [14, 23]. In this model, functional properties correspond to duties and non-functional properties correspond to performance metrics about these duties. We consider 3 duties as per Fig. 5; a thing *senses* the cyber-physical surrounding so that it generates data; a thing *actuates* data including those that are sensed; and a thing *communicates* with the cyber-physical surrounding the data that are sensed and/or actuated. To keep the paper focussed on BP-IoT collaboration, we refer readers to [23] for more details about things' QoT non-functional properties.
  In conjunction with adopting the QoT model, we comply with the Web of Things (WoT) Thing Description[6]

[6] www.w3.org/TR/wot-thing-description.

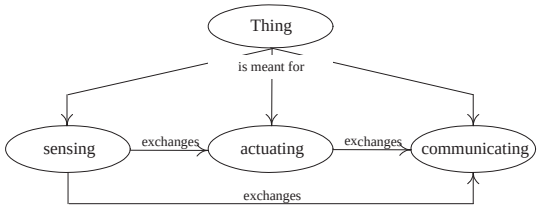


Figure 5: Thing's functional properties as duties

to describe things. Listing 1 is an example of a moisture-sensor description in JSON-LD where lines 2-3 refer to semantic metadata, lines 4-6 refer to details about the thing, lines 7-17 refer to interaction resources, lines 18-21 refer to communication, lines 22-26 refer to security, and, finally, lines 27-57 refer to QoT properties.

Listing 1: Moisture sensor's WoT CP description

```
1  {
2          "@context": ["https://w3c.github.io/wot/w3c-wot-
               td-context.jsonld",
3          "https://w3c.github.io/wot/w3c-wot-common-
               context.jsonld"],
4          "@type": ["Sensor"],
5          "name": "myMoistureSensor",
6          "base": "coap:///www.example.com:5683/moisture-
               sensor/",
7          "interaction": [{
8                  "@type": ["Property","Moisture"],
9                  "name": "MoistureSensor",
10                 "schema":{ "type": "number" },
11                 "writable": false,
12                 "observable":true,
13                 "form": [{
14                         "href": "val",
15                         "mediaType": "application/json"
16                         }],
17         }],
18         "link": [{
19                 "href": "coap://moisture.example.com:568
                       3/ev",
20                 "mediaType": "application/json"
21         }],
22         "security": {
23                 "cat": "token:jwt",
24                 "alg": "HS256",
25                 "as": "https://authority-issuing.example
                       .org"
26         },
27         "quality":[{
28                         "type":"sensing",
29                         "name":"Frequency of sensing",
30                         "property":"frequency",
31                         "value":"continuous"
32                 },{
33                         "type":"sensing",
34                         "name":"Quality of sensed
                               outcome",
35                         "property":"outcomequality",
36                         "value":"high"
37                 },{
38                         "type":"sensing",
39                         "name":"Resource",
40                         "property":["resource","energy"]
                               ,
41                         "value":"high"
42                 },{
43                         "type":"communicating",
44                         "name":"Reception rate of sensed
                                outcome",
45                         "property":["reception","
                               bandwidth"],
46                         "value":"high"
47                 },{
48                         "type":"communicating",
49                         "name":"Delivery rate of sensed
                               outcome",
50                         "property":["deliveryrate","
                               bandwidth"],
51                         "value":"high"
52                 },{
53                         "type":"reasoning",
```

```
                        "name":"Accuracy of decision",
                        "property":"accuracy",
                        "value":"high"
                }]
}
```

- *BP-transfer* link "ships" the descriptions of process models to PoTs by involving the BPMS's exchange module as a sender and the IoTMS's exchange module as a recipient. Upon description receipt, the IoTMS's connection functionality process this description so that details about process models like dependencies between things and input/output messages are extracted. The objective of analyzing these details is to establish the flow of things like shown in Fig. 4.
  In the BP-IoT framework, we formalize a process model as a *5-tuple* $PM = < \mathcal{A}, \mathcal{L}, \mathcal{T}r, s_0, \mathcal{F} >$ *where:* $\mathcal{A}$ *is a finite set of activity names;* $\mathcal{L}$ *is a set of labels;* $\mathcal{T}r \subseteq \mathcal{A} \times \mathcal{L} \times \mathcal{A}$ *is the transition relation; each transition* $tr = (a_{sr}, l, a_{tg})$ *consists of a source activity* $a_{sr} \in \mathcal{A}$*, a target state* $a_{tg} \in \mathcal{A}$*, and a transition label* $l \in \mathcal{L}$*;* $A_0 \in \mathcal{S}$ *is the initial activity; and* $\mathcal{F} \subseteq \mathcal{S}$ *is a set of final activities*. In line with using JSON-LD to describe things (Listing 1) we will examine the conversion of BPMN-based process models into JSON.

Below we present *interaction*.

- *BP-interaction* supports the invocation of things that are included in TaPs' process models. To this end the BPMS's accessibility functionality identifies (activity,thing) couples and thus, the commands that the activities will execute over things.
- *T-interaction* supports the invocation of things that are included in PoT' process models. To this end the IoTMS's connection functionality identifies (activity,thing) couples and thus, the commands that activities will execute over things.

5 SYSTEM IMPLEMENTATION

To demonstrate the technical doability of the BP-IoT framework, we developed a Web-based system for TaP generation. Its architecture is given in Fig 6.

The TaP generator implementation is in line with our proposed framework in term of concern separation (Fig. 2) and is put into action as follows. First, a BP designer draws a BP's process model using the design module based on predefined activities stored in the repository of activities. This module uses bpmn-js (bpmn.io/toolkit/bpmn-js), a BPMN 2.0 rendering toolkit and Web modeler, for designing BPs that uses a dedicated palette and generates the associated BPMN models. In this step, the necessary BP activities are retrieved from the repository of activities. The obtained BPMN models are then stored in the repository of BPs. Once a BP's process model is created, it can be retrieved and edited using the design module. Afterwards the BP designer associates things with adequate activities. To this end, the BP designer selects an activity from the designed BP's process model and chooses an adequate thing from a list of things submitted by the exchange module developed in Angular 7 framework (angular.io). For this experiment we populated a repository of configured things with several things' descriptions defined in WoT TD model (Listing 1). Finally, the BP model and selected things $t_{1..j}$ are transmitted to the binding module implemented in JavaScript NodeJS (nodejs.org) platform which generates the associated TaP (Algorithm 1) and stores it in the repository of TaPs. All repositories are implemented as tables in a MongoDB database and accessed using mongoose library (mongoosejs.com). Fig. 7 is a screenshot of binding things to activities after completing the design of the BP that includes these activities.

6 CONCLUSION

This paper presented a framework for enacting and supporting the collaboration of BPs and IoT. This collaboration is deemed necessary because of the limitations that whether BPs and IoT run-into and hence, could impact their competitiveness among other ICT. The framework design complied with the separation-of-concerns principle since it is spread over 2 worlds, cyber and physical, each comprising necessary repositories and modules. How can BP support IoT and how can IoT support BP are the main concerns that shaped the design and development of our BP-IoT collaboration framework. In term of future work, we would like to demonstrate the implementation of PoTs along with testing both TaPs and PoTs and using formal techniques like model checking to ensure the consistency of obtained TaPs and PoTs.

REFERENCES

[1] D. Andročec, B. Tomaš, and T. Kišasondi. Interoperability and Lightweight Security for Simple IoT Devices. In *Proceedings of the Information Systems Security Conference (ISS'2017) held in conjunction with the 40th Jubilee International Convention on Information and Communication Technology, Electronics, and Microelectronics (MIPRO'2017)*, Opatija, Croatia, May 2017.

[2] A. Bröring, A. Ziller, V. Charpenay, S. Schmid, A. Thuluva, D. Anicic, A. Zappa, M. Linares, L. Mikkelsen, and C. Seidel. The BIG IoT API - Semantically Enabling IoT Interoperability. *IEEE Pervasive Computing*, August 2018 (forthcoming).

[3] N. Brouns, S. Tata, H. Ludwig, E. Asensio, and P. Grefen. Modeling IoT-aware Business Processes: A State of the Art Report. Technical Report IBM Research Report RJ10540, 2018.

[4] Y. Cheng, S. Zhao, B. Cheng, and J. Chen. A Resource Oriented Modeling Approach for the Internet of Things: A Business Process Perspective. In *Proceedings of MODELS'2017 Satellite Event held in conjunction with ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS'2017)*, Austin, TX, USA, 2017.

[5] S. Cherrier and V. Deshpande. From BPM to IoT. In *Business Process Management Workshops held in conjunction with BPM'2017*, Barcelona, Spain, 2017.

[6] M. Dastani, L. van der Torre, and N. Yorke-Smith. Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (Fortino, G. and Russo, W. and Savaglio, C. and Shen, W. and Zhou, M.), 2017 (forthcoming).

[7] D. Domingos and F. Martins. Using BPMN to Model Internet of Things Behavior within Business Process. *International Journal of Information Systems and Project Management*, 5(4), 2017.

[8] DZone. The Internet of Things, Application, Protocls, and Best Practices. Technical report, https://dzone.com/guides/iot-applications-protocols-and-best-practices, 2017 (visited in May 2017).

[9] C. Friedow, M. Völker, and M. Hewelt. Integrating IoT Devices into Business Processes. In *Proceedings of the 1st Workshop on*

Algorithm 1 TaP formation algorithm

Input Set of activities $\mathcal{A}\langle a_1, ..., a_i\rangle$ and Repository of things $\mathcal{R}_\mathcal{T}$
Output Set of (activity, thing): $\langle(a_1, t_1), (a_2, null), ..., (a_i, t_i)\rangle$
▷ *null*: not all activities connect to things
for all $a_i \in \mathcal{A}$ **do**
  if $bind(a_i)$=true **then** ▷ *bind()* checks if an activity is linked to a thing
    $consult(\mathcal{R}_\mathcal{T})$
    if $\exists\ t_j \in \mathcal{R}_\mathcal{T}$ s.t $match(a_i,\ T\text{-}transfer(\text{duties}(t_j)))$=true **then**
      ▷ BPMS's context functionality identifies a thing's duties after transferring its description
      ▷ *match()* checks if a thing's duties satisfy an activity's needs
      $confirm(a_i,\ BP\text{-}interaction(t_j))$
      ▷ BPMS's accessibility functionality confirms the connection(activity,thing)
    else stop TaP formation
    end if
  end if
end for
return $\langle(a_i, t_i|null)\rangle$

Algorithm 2 PoT formation algorithm

Input BP's process model $\mathcal{PM}$< $\mathcal{A}$, $\mathcal{L}$, $\mathcal{T}r$, s_0, $\mathcal{F}$ > and Set of things $\mathcal{T}\langle t_1, ..., t_j\rangle$
Output PoT's process model $\mathcal{TPM}$< $\mathcal{T}$, $\mathcal{L}$, $\mathcal{T}r$, s_0, $\mathcal{F}$ >
▷ revised process model where activities are associated with things
PoT_{flow}=$BP\text{-}transfer(\mathcal{PM})$
▷ IoTMS's accessibility functionality extracts details from $\mathcal{PM}$
for all $a_i \in \text{PoT}_{flow}$ **do** ▷ a_i is an activity in $\mathcal{PM}$
  $confirm(t_i,\ T\text{-}interaction(a_i))$
  ▷ IoTMS's connection functionality confirms (thing,activity) association
end for
for all $tr_i \in \text{PoT}_{flow}$ **do** ▷ tr_i is a transition in $\mathcal{PM}$
  $\mathcal{TPM} = \mathcal{TPM}.append(connect(t_i,\ tr_i,\ t_{i+1}))$
  ▷ IoTMS's connection functionality establishes the dependency between things
end for
return $\mathcal{TPM}$

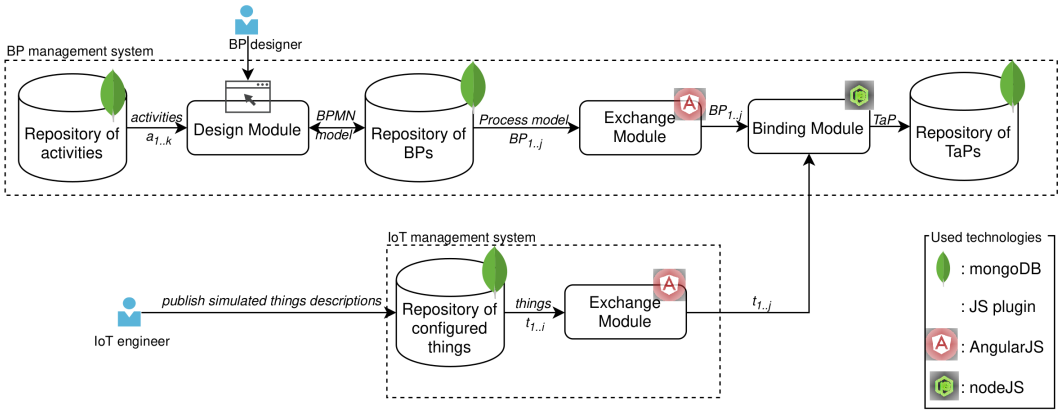


Figure 6: Architecture of TaP generator

Flexible Advanced Information Systems (FAiSE'2018) held in conjunction with CAiSE'2018, Tallinn, Estonia, 2018.
[10] H. Green. The Internet of Things in the Cognitive Era: Realizing the Future and Full Potential of Connected Devices. www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WWW12366USEN, December 2015.
[11] P. Grefen, N. Brouns, L. Ludwig, and E. Serral. Co-location Specification for IoT-Aware Collaborative Business Processes. In *Proceedings of CAiSE'2019 Forum held in conjunction with the*

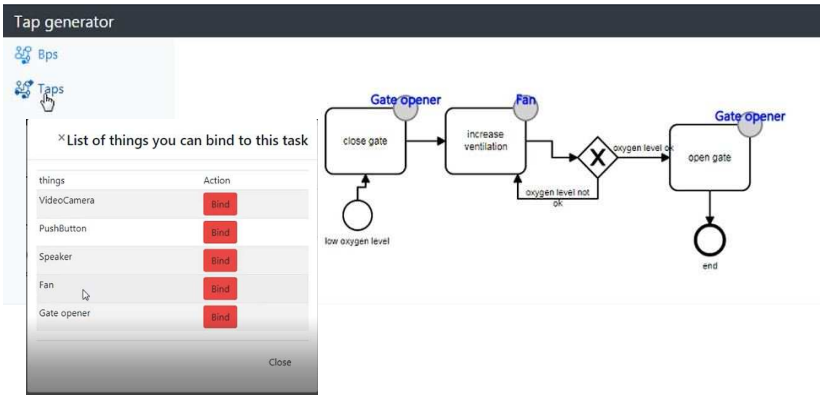


Figure 7: Screenshot of binding things to activities

31st International Conference on Advanced Information Systems Engineering (CAiSE'2019), Rome, Italy, 2019.

[12] F. Hasić and E. Serral Asensio. Executing IoT Processes in BPMN 2.0: Current Support and Remaining Challenges. In *Proceedings of the IEEE 13th International Conference on Research Challenges in Information Science (RCIS'2019)*, Brussels, Belgium, May 2019.

[13] Z. Maamar, T. Baker, N. Faci, E. Ugljanin, Y. Atif, M. Al-Khafajiy, and M. Sellami. Cognitive Computing Meets the Internet of Things. In *Proceedings of the 13th International Conference on Software Technologies (ICSOFT'2018)*, Porto, Portugal, 2018.

[14] Z. Maamar, T. Baker, M. Sellami, M. Asim, E. Ugljanin, and N. Faci. Cloud *versus* Edge: Who Serves the Internet-of-Things Better? *Internet Technology Letters, Wiley*, 1(5), June 2018.

[15] Z. Maamar, K. Boukadi, E. Ugljanin, T. Baker, M. Asim, M. Al-Khafajiy, D. Benslimane, and H. El Alaoui El Abdallaoui. Thing Federation as a Service: Foundations and Demonstration. In *Proceedings of the 8th International Conference on Model and Data Engineering (MEDI'2018)*, Marrakesh, Morocco, 2018.

[16] Z. Maamar, M. Sellami, N. Faci, E. Ugljanin, and Q. Sheng. Storytelling Integration of the Internet of Things into Business Processes. In *Proceedings of the Business Process Management Forum (BPM Forum'2018) held in conjunction with the 16th International Conference on Business Process Management (BPM'2018)*, Sydney, NSW, Australia, 2018.

[17] S. Mandal, M. Hewelt, M. Oestreich, and M. Weske. A Classification Framework for IoT Scenarios. In *Proceedings of the 2nd International Workshop on BP-Meet-IoT held conjunction with BPM'2018*, Sydney, Australia, September 2018.

[18] D. Marlon, L. R. Marcello, M. Jan, and H. Reijers. *Fundamentals of Business Process Management*. Springer, 2013.

[19] F. Martins and D. Domingos. Modelling IoT Behaviour within BPMN Business Processes. In *Proceedings of the International Conference on ENTERprise Information Systems (CENTERIS'2017)*, Barcelona, Spain, 2017.

[20] G. Meroni. Integrating the Internet of Things with Business Process Management: A Process-aware Framework for Smart Objects. In *Proceedings of CAiSE'2015 Doctoral Consortium at the 27th International Conference on Advanced Information Systems Engineering (CAiSE'2015)*, Stockholm, Sweden, 2015. Springer Link.

[21] S. Meyer, A. Ruppen, and C. Magerkurth. Internet of Things-Aware Process Modeling: Integrating IoT Devices as Business Process Resources. In *Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE'2013)*, Valencia, Spain, 2013.

[22] D. Mukherjee, D. Pal, and P. Misra. Workflow for the Internet of Thing. In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS'2017)*, Porto, Portugal, 2017.

[23] A. Qamar, A. Muhammad, Z. Maamar, T. Baker, and S. Saeed. A Quality-of-Things Model for Assessing the Internet-of-Thing's Non-Functional Properties. *Transactions on Emerging Telecommunications Technologies*, 2019 (forthcoming).

[24] M. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke. Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 3(1), 2016.

[25] S. Schönig, L. Ackermann, and S. Jablonski. Internet of Things Meets BPM: A Conceptual Integration Framework. In *Proceedings of the 8th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH'2018)*, Porto, Portugal, 2018.

[26] S. Schönig, L. Ackermann, S. Jablonski, and A. Ermer. An Integrated Architecture for IoT-Aware Business Process Execution. In *Proceedings of the 19th Edition of Business Process Modeling, Design and Support Conference (BPMDS'2018) held in conjunction with CAiSE'2018*, Tallinn, Estonia, 2018.

[27] R. Seiger, M. Gohlke, and U. Aßmann. Augmented Reality-Based Process Modelling for the Internet of Things with HoloFlows. In *Proceedings of the 20th International Conference on Enterprise, Business-Process and Information Systems Modeling (BPMDS'2019) held in conjunction with CAiSE'2019*, Rome, Italy, 2019.

[28] R. Seiger, P. Heisig, and U. Aβmann. Retrofitting of Workflow Management Systems with Self-X Capabilities for Internet of Things. In *Proceedings of the 2nd International Workshop on BP-Meet-IoT held conjunction with BPM'2018*, Sydney, Australia, September 2018.

[29] Z. Sheng, C. Mahapatra, C. Zhu, and V. Leung. Recent Advances in Industrial Wireless Sensor Networks Toward Efficient Management in IoT. *IEEE Access*, 3, 2015.

[30] K. Suri, W. Gaaloul, A. Cuccuru, and S. Gerard. Semantic Framework for Internet of Things-Aware Business Process Development. In *Proceedings of the 26th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'2017)*, Poznan, Poland, 2017.

[31] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A. Biswas, and K. Moessner. Enabling Smart Cities through a Cognitive Management Framework for the Internet of Things. *IEEE Communications Magazine*, 51(6), 2013.

[32] S. Ware, R. Young, B. Harrison, and D. Roberts. A computational model of narrative conflict at the fabula level. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, 6(3), 2014.

[33] M. Weiser. The Computer for the 21^{st} Century. *Newsletter ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3), 199.

[34] M. Weske. *Business Process Management - Concepts, Languages, Architectures*. 2nd Edition,Springer, 2012.

[35] Q. Wu, G. Ding, Y. Xu, S. Feg, Z. Du, J. Wang, and K. Long. Cognitive Internet of Things: A New Paradigm Beyond Connection. *IEEE Internet of Things Journal*, 1(2), April 2014.

[36] F. Zambonelli. Key Abstractions for IoT-Oriented Software Engineering. *IEEE Software*, 34(1), January-February 2017.

Consistent Query Answering with Prioritized Active Integrity Constraints

Marco Calautti
University of Trento
Trento (TN) - Italy
marco.calautti@unitn.it

Luciano Caroprese
ICAR-CNR
Rende (CS) - Italy
l.caroprese@dimes.unical.it

Sergio Greco
University of Calabria
Rende (CS) - Italy
greco@dimes.unical.it

Cristian Molinaro
University of Calabria
Rende (CS) - Italy
cmolinaro@dimes.unical.it

Irina Trubitsyna
University of Calabria
Rende (CS) - Italy
trubitsyna@dimes.unical.it

Ester Zumpano
University of Calabria
Rende (CS) - Italy
e.zumpano@dimes.unical.it

ABSTRACT

Consistent query answering is a principled approach for querying inconsistent databases. It relies on two basic notions: the notion of a *repair*, that is, a consistent database that "minimally" differs from the original one, and the notion of a *consistent query answer*, that is, a query answer that can be derived from every repair. In general, an inconsistent database can admit multiple repairs, each corresponding to a different way of restoring consistency, and the consistent query answering framework does not make any discrimination among them. However, in many applications it is natural and desired to express preferences among the different choices that can be made to resolve inconsistency.

In this paper, we consider the framework of *Prioritized Active Integrity Constraints* (PAICs), a declarative and powerful form of active rules which enable users to express a wide range of integrity constraints along with preferences on how consistency should be restored. PAICs induce preferences among repairs, so that a set of "preferred" ones can be identified. Then, "preferred" query answers are naturally defined as query answers derived from preferred repairs only.

We show how preferred repairs can be obtained from the preferred stable models of a prioritized logic program derived from a given a set of PAICs. Furthermore, we study the restricted class of *Prioritized Active Functional Dependencies* (PAFDs), which admits a unique preferred repair and for which query answering can be accomplished in polynomial time.

CCS CONCEPTS

• Information systems → Integrity checking; **Database query processing**; **• Theory of computation → Constraint and logic programming**; **Logic and databases**; **Database query processing and optimization (theory)**.

KEYWORDS

Active Integrity Constraints, Database Repairs, Consistent Query Answering

ACM Reference Format:

Marco Calautti, Luciano Caroprese, Sergio Greco, Cristian Molinaro, Irina Trubitsyna, and Ester Zumpano. 2020. Consistent Query Answering with Prioritized Active Integrity Constraints. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410592

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410592

1 INTRODUCTION

Integrity constraints have long been used to maintain database consistency, thus ensuring that a database reflects a valid, consistent state of the world. However, nowadays several applications have to deal with *inconsistent* databases, namely databases which violate given integrity constraints, because integrity constraints may not be enforced or satisfied. For instance, inconsistency may arise in *data integration*, where multiple autonomous sources are integrated together. Even if the sources are separately consistent, the integrated database may be inconsistent. Inconsistency may also occur when integrity constraints are unenforced because integrity checking is infeasible or too costly. There are plenty of other scenarios where inconsistency arises.

Dealing with inconsistent databases we face the problem of extracting reliable information from them. In this regard, most of the works in the literature are based on the *Consistent Query Answering* (CQA) framework. This framework relies on the notions of *repair* and *consistent query answer*. Intuitively, a repair for a possibly inconsistent database is a consistent database which "minimally" differs from the original one. In general, there may be more than one repair for an inconsistent database. The *consistent answers* to a query over a possibly inconsistent database are those query answers that can be obtained from every repair. The following example illustrates these notions.

Example 1.1. Consider a database consisting of two relations with schemas $dept(Name)$ and $emp(Name, Dept)$, where the former stores information on departments and the latter about employees and the departments they work for. Suppose a referential integrity constraint is defined, stating that every department appearing in the *emp* relation must occur in the *dept* relation too. This constraint can be defined through the following first-order formula:

$$\forall \mathtt{E}\, \forall \mathtt{D}\ [\ \mathtt{emp(E,D)} \Rightarrow \mathtt{dept(D)}\ ]$$

Consider now the inconsistent database $D = \{\mathtt{emp(john, cs)}, \mathtt{emp(john, dimes)}, \mathtt{dept(dimes)}\}$. A repair can be obtained by applying a "minimal" set of update operations (we consider fact insertions or deletions only) to the original database. Specifically, there exist two possible repairs: R_1 obtained by inserting the fact $\mathtt{dept(cs)}$, and R_2 obtained by deleting the fact $\mathtt{emp(john, cs)}$. The only consistent query answer to the query asking for all departments' names is dimes. □

The motivation of this work stems from the observation that an inconsistent database can be repaired in different ways, and in many applications it is natural and desirable to express preferences among the different actions that make the database consistent.

Example 1.2. Consider again the database of Example 1.1. Recall that it can be repaired by either inserting $\mathtt{dept(cs)}$ or by deleting $\mathtt{emp(john, cs)}$. In this scenario, suppose that the insertion of a missing department is preferable to the deletion of an existing employee. This preference can be expressed by means of the following prioritized active integrity constraint:

$$\forall \mathtt{E}\, \forall \mathtt{D}\ [\ \mathtt{emp(E,D)} \wedge \mathtt{not\ dept(D)}\ \Rightarrow\ \mathtt{+dept(D)} \succ \mathtt{-emp(E,D)}\ ]$$

As R_1 is obtained by inserting $\mathtt{dept(cs)}$ whereas R_2 deletes $\mathtt{emp(john, cs)}$, then R_1 *is preferable to* R_2. Therefore, we have a unique preferred repair and the preferred consistent answers to the query of Example 1.1 are both departments dimes and cs. □

In this paper we consider *Prioritized Active Integrity Constraints* (PAICs) [17], a declarative and powerful form of active rules, enabling users to express both integrity constraints and preferences on how inconsistency should be resolved.

We first show the approach for computing preferred repairing update sets. In particular, our approach exploits a rewriting to prioritized logic programs, in such a way that the stable models of the program coincide with the preferred repairing update sets.

Then, we focus on an interesting restricted class of PAICs, called *Prioritized Active Functional Dependencies* (PAFDs), which allows us to express functional dependencies, and enjoys important properties that are of relevance in practice: they admit a unique preferred repairing update set, and consistent query answers can be computed in polynomial time.

The rest of the paper is organized as follows. Preliminaries are reported in Section 2. Section 3 recalls Prioritized Active Integrity Constraints. The computation of preferred repairing update sets is addressed in Section 4. The special case of Prioritized Active Functional Dependencies is studied in Section 5. Related work is discussed in Section 6. Conclusions are drawn in Section 7.

2 PRELIMINARIES

In this section, we provide preliminaries on answer set programming, prioritized logic programming, and (active) integrity constraints.

We start by introducing common notation and terminology. We assume the existence of the following (pairwise disjoint) enumerable sets: a set of *predicates* $\mathcal{Q}$, a set of *variables* $\mathcal{V}$, and a set of *constants* C. Each predicate is associated with an *arity*, which is a non-negative integer. A *term* is either a variable or a constant. An *atom* is of the form $p(t_1, \ldots, t_n)$, where p is a predicate of arity n and the t_i's are terms—when the atom's predicate is relevant, we call such an atom a p-atom. We write an atom also as $p(\bar{t})$, where $\bar{t}$ is understood to be a sequence of terms, and write $p(\bar{X})$ (resp. $p(\bar{c})$) when all terms are variables (resp. constants). A *literal* is either an atom A (*positive* literal) or its negation $\neg A$ (*negative* literal).

2.1 Answer Set Programming

Syntax. A *rule* r is of the form:

$$A_1 \vee \cdots \vee A_m \ \leftarrow\ B_1 \wedge \cdots \wedge B_k \wedge \neg C_1 \wedge \cdots \wedge \neg C_n \wedge \varphi$$

where $m > 0,\ k \geq 0,\ n \geq 0, A_1, \ldots, A_m, B_1, \ldots, B_k, C_1, \ldots, C_n$ are atoms, and φ is a conjunction of *comparison atoms* of the form $t_1 \theta t_2$, where t_1 and t_2 are terms, and $\theta \in \{=, \neq, >, \geq, <, \leq\}$. The disjunction on the left-hand side of $\leftarrow$ is called the *head* of r and is denoted by $head(r)$. The conjunction on the right-hand side of $\leftarrow$ is called the *body* of r and is denoted by $body(r)$. If $m = 1$, then r is *normal*; in this case, $head(r)$ denotes the head atom. If $n = 0$, then r is *positive*.

A *program* is a finite set of rules. A program is *normal* (resp. *positive*) if every rule in it is normal (resp. positive). We assume that programs are *safe*, i.e., for every rule, every variable appears in some positive body literal (i.e., the B_i's in the rule above). An atom (resp. literal, rule, program) is *ground* if no variables occur in it. A ground normal rule with an empty body is also called a *fact*—we write a fact simply as A rather than $A \leftarrow$.

Semantics. Consider a program P. The *Herbrand universe* U_P of P is the set of all constants appearing in P. The *Herbrand base* B_P of P is the set of all ground atoms that can be built using predicates appearing in P and constants in U_P.

A *substitution* θ is of the form $\{X_1/t_1, \ldots, X_n/t_n\}$, where $X_1, \ldots, X_n$ are distinct variables and $t_1, \ldots, t_n$ are terms. A substitution where all t_i's are constants is called *matcher*. The result of applying θ to an atom A, denoted $A\theta$, is the atom obtained from A by simultaneously replacing each occurrence of a variable X_i in A with t_i if X_i/t_i belongs to θ.

A rule (resp. atom) r' is a *ground instance* of a rule (resp. atom) r in P if r' can be obtained from r by substituting every variable in r with some constant in U_P. We use $ground(r)$ to denote the set of all ground instances of r and $ground(P)$ to denote the set of all ground instances of the rules in P, i.e., $ground(P) = \bigcup_{r \in P} ground(r)$.

An *interpretation* of P is any subset I of B_P. The truth value of a ground atom A w.r.t. I, denoted $value_I(A)$, is *true* if $A \in I$, *false* otherwise. The truth value of $\neg A$ w.r.t. I, denoted $value_I(\neg A)$, is *true* if $A \notin I$, *false* otherwise. A ground rule r is *satisfied* by I, denoted

$I \models r$, if there is a ground literal L in $body(r)$ s.t. $value_I(L) = false$, or some comparison atom in $body(r)$ is false, or there is a ground atom A in $head(r)$ s.t. $value_I(A) = true$. Thus, if the body of r is empty, r is satisfied by I if there is an atom A in $head(r)$ s.t. $value_I(A) = true$.

An interpretation of P is a *model* of P if it satisfies every ground rule in $ground(P)$. A model M of P is minimal if no proper subset of M is a model of P. The set of all minimal models of P is denoted by $MM(P)$. Given an interpretation I of P, let P^I denote the ground positive program derived from $ground(P)$ by *(i)* removing every rule containing a negative literal $\neg A$ in the body with $A \in I$, and *(ii)* removing all negative literals from the remaining rules. An interpretation I is a *stable model* of P if $I \in MM(P^I)$. The set of stable models of P is denoted by $SM(P)$. It is well known that $SM(P) \subseteq MM(P)$, and $SM(P) = MM(P)$ for positive programs.

We will also allow rules of the form $\leftarrow\ L_1 \wedge \cdots \wedge L_m$ called *denial rules* which are satisfied if the body is false and can be seen as a shorthand for the recursive rule $p \leftarrow\ L_1 \wedge \cdots \wedge L_m \wedge \neg p$, where p is a fresh predicate of arity zero, not appearing elsewhere. Denial rules are used to force the conjunction $L_1 \wedge \cdots \wedge L_m$ to be false in every stable model.

2.2 Prioritized Logic Programming

Several authors have investigated various forms of priorities into logic languages [9, 33, 54, 62]. In this paper, we refer to the extension proposed in [62], which is recalled below.

Syntax. A *priority* is a statement of the form $A_1 \succeq A_2$, where A_1 and A_2 are ground atoms. If $A_1 \succeq A_2$, then we say that A_1 *has a higher priority than* A_2. We write $A_1 \succ A_2$ if $A_1 \succeq A_2$ and $A_2 \not\succeq A_1$. Given two non-ground atoms A_1 and A_2, then $A_1 \succeq A_2$ stands for the set of all priorities $A_1' \succeq A_2'$ where A_1' and A_2' are ground instances of A_1 and A_2, respectively. Given a set Φ of priorities, then Φ^* denotes the set of priorities that can be reflexively or transitively derived from Φ.

A *prioritized logic program* (PLP) is a pair (P, Φ), where P is a program and Φ is a set of priorities over B_P.

Semantics. Given a prioritized logic program (P, Φ), the relation $\sqsupseteq$ is defined over the stable models of P as follows. For any stable models M_1, M_2, and M_3 of P:

- $M_1 \sqsupseteq M_1$.
- $M_1 \sqsupseteq M_2$ if there exists $A_1 \in M_1 \setminus M_2$ such that:
  - $\exists A_2 \in M_2 \setminus M_1$ such that $(A_1 \succeq A_2) \in \Phi^*$, and
  - $\nexists\, A_3 \in M_2 \setminus M_1$ such that $(A_3 \succ A_1) \in \Phi^*$.
- If $M_1 \sqsupseteq M_2$ and $M_2 \sqsupseteq M_3$, then $M_1 \sqsupseteq M_3$.

If $M_1 \sqsupseteq M_2$, then we say that M_1 is *preferable* to M_2 w.r.t. Φ. Moreover, we write $M_1 \sqsupset M_2$ if $M_1 \sqsupseteq M_2$ and $M_2 \not\sqsupseteq M_1$.

Observe that the application of the definition above presumes first the direct comparison of models on the base of their atoms (first two bullets), and next the application of the transitive property (third bullet). The distinct models M_1 and M_2 are compared directly if they have comparable atoms A_1 and A_2 respectively, and do not share these atoms. In particular, M_1 is directly preferable to M_2 if $(A_1 \succeq A_2) \in \Phi^*$ and there is no atom A_3 with a strongly higher priority than A_1, i.e. $(A_3 \succ A_1) \in \Phi^*$, occurring in $M_2 \setminus M_1$.

A *preferred stable model* of a PLP (P, Φ) is a stable model M of P such that there is no stable model M' of P such that $M' \sqsupset M$. The set of all preferred stable models of (P, Φ) is denoted by $PSM(P, \Phi)$.

2.3 Databases and Integrity Constraints

A *database* is a finite set of facts. We consider queries expressed by means of nonrecursive safe normal programs which are equivalent to relational algebra (RA) or to safe relational calculus (SRC). Thus, a *query* Q is a pair $\langle P, g\rangle$, where P is a nonrecursive safe normal program (or a RA expression, or a SRC formula) and g is a predicate appearing in P. The result of evaluating Q on a database D, denoted $Q(D)$, is the set of all g-atoms occurring in the (unique) stable model of $D \cup Q$.

An *(universally quantified or full) integrity constraint* (IC) is of the form:

$$\forall \overline{X} \left[\varphi(\overline{X}_0) \wedge \bigwedge_{i=1}^{m} b_i(\overline{X}_i) \Rightarrow \bigvee_{i=m+1}^{n} b_i(\overline{X}_i) \right]$$

where $n \geq m \geq 0$, $\overline{X} = \bigcup_{i=1}^{m} \overline{X}_i$, $\overline{X}_i \subseteq \overline{X}$ for every $0 \leq i \leq n$, $\varphi(\overline{X}_0)$ is a conjunction of comparison atoms, and the $b_i(\overline{X}_i)$'s are atoms. W.l.o.g. we do not allow constants in the $b_i(\overline{X}_i)$'s, as each constant c in such atoms can be removed by replacing it with a fresh variable X_j and adding $X_j = c$ to the conjunction $\varphi(\overline{X}_0)$.

Notice that, similarly to the safety condition defined for rules, every variable in an integrity constraint that appears in $\varphi(\overline{X}_0)$ or in the head must appear in some (standard) atom in the body. The integrity constraint above can also be written as follows:

$$\forall \overline{X} \left[\varphi(\overline{X}_0) \wedge \bigwedge_{i=1}^{m} b_i(\overline{X}_i) \wedge \bigwedge_{i=m+1}^{n} not\ b_i(\overline{X}_i) \ \Rightarrow \right]$$

A database D is *consistent* w.r.t. a set of integrity constraints Σ (we also say that D *satisfies* Σ) if $D \models \Sigma$ in the standard model-theoretic sense. Otherwise, D is *inconsistent*.

In the following, every set of integrity constraints Σ is assumed to be satisfiable, that is, there exists a database satisfying Σ. For instance, constraints of the form above with $m > 0$ are satisfied by the empty database.

A (ground) *update atom* is of the form $+a(\overline{X})$ or $-a(\overline{X})$, where $a(\overline{X})$ is a (ground) atom. Intuitively, a ground update atom $+a(\overline{c})$ states that $a(\overline{c})$ will be inserted into the database, whereas a ground update atom $-a(\overline{c})$ states that $a(\overline{c})$ will be deleted from the database. We sometimes use the notation $\pm a(\overline{X})$ to refer to an update atom, meaning that it is either $+a(\overline{X})$ or $-a(\overline{X})$.

The *complementary literal* of an update atom $+a(\overline{X})$ (resp. $-a(\overline{X})$) is $cLit\ (+a(\overline{X})) = not\ a(\overline{X})$ (resp. $cLit(-a(\overline{X})) = a(\overline{X})$). Similarly, $cUpd(+a\ (\overline{X})) = -a(\overline{X})$ (resp. $cUpd(-a(\overline{X})) = +a(\overline{X})$) denotes the "complementary" update atom of $+a(\overline{X})$ (resp. $-a(\overline{X})$).

Given a set U of ground update atoms, the following sets are defined:

$$U^+ = \{a(\overline{c}) \mid +a(\overline{c}) \in U\},$$
$$U^- = \{a(\overline{c}) \mid -a(\overline{c}) \in U\},$$
$$cLit(U) = \{cLit(\pm a(\overline{c})) \mid \pm a(\overline{c}) \in U\},$$
$$cUpd(U) = \{cUpd(\pm a(\overline{c})) \mid \pm a(\overline{c}) \in U\}.$$

We say that U is *coherent* if it does not contain two update atoms $+a(\overline{c})$ and $-a(\overline{c})$ (i.e., if $U^+ \cap U^- = \emptyset$). Given a database D and a coherent set of update atoms U, we use $U(D)$ to denote the database obtained by applying U to D, that is, the database $(D \cup U^+) \setminus U^-$.

Given a database D and a set of integrity constraints Σ, a *repairing update set* (RUS) for $\langle D, \Sigma\rangle$ is a coherent set of ground update atoms R such that *(i)* $R(D) \models \Sigma$ and *(ii)* there is no proper subset R' of R such that $R'(D) \models \Sigma$. The set of all possible repairing update sets for $\langle D, \Sigma\rangle$ is denoted as $\mathbf{R}(D, \Sigma)$. Every database obtained by applying a repairing update set R to D is called a *repair* for $\langle D, \Sigma\rangle$. Thus, repairs are consistent databases derived from the original one by means of a minimal set of update operations. We consider fact insertions and deletions as the only primitives to restore consistency.

The *certain (or consistent) answers* to a query Q on a database D w.r.t. a set of integrity constraints Σ are defined as

$$\textsc{certain}(Q, D, \Sigma) = \bigcap_{R \in \mathbf{R}(D,\Sigma)} Q(R(D)),$$

whereas the *possible answers* are defined as

$$\textsc{possible}(Q, D, \Sigma) = \bigcup_{R \in \mathbf{R}(D,\Sigma)} Q(R(D)).$$

A general approach for the computation of repairing update sets and consistent query answers has been proposed in [44, 55]. The technique is based on the generation of a program $DP(\Sigma)$ derived from a set of integrity constraints Σ, so that the repairing update sets for a database D can be derived from the stable models of $DP(\Sigma) \cup D$.

2.4 Active Integrity Constraints

We now recall the syntax and semantics of *active integrity constraints* (AICs) [18, 19]. In the next section, we will see that prioritized active integrity constraints generalize AICs by allowing preferences to be expressed among update atoms.

Syntax. Integrity constraints do not allow us to specify which actions can or cannot be performed to resolve inconsistency. AICs overcome this limitation by allowing users to express which actions are admitted to restore consistency.

Definition 2.1. An *active integrity constraint* (AIC) is of the form:

$$\forall \overline{X} \left[\varphi(\overline{X}_0) \wedge \bigwedge_{i=1}^{m} b_i(\overline{X}_i) \wedge \bigwedge_{i=m+1}^{n} not\ b_i(\overline{X}_i) \Rightarrow \bigvee_{i=n+1}^{p} \pm a_j(\overline{X}_j) \right] \quad (1)$$

where $\overline{X} = \bigcup_{i=1}^{m} \overline{X}_i$, $\overline{X}_i \subseteq \overline{X}$ for every $0 \leq i \leq p$, $\varphi(\overline{X}_0)$ is a conjunction of comparison atoms, the $b_i(\overline{X}_i)$'s are atoms, and the $\pm a_i(\overline{X}_i)$'s are update atoms. □

For an AIC σ, the conjunction (resp. disjunction) on the left-hand (resp. right-hand) side of $\Rightarrow$ is called the *body* (resp. *head*) of σ, denoted $body(\sigma)$ (resp. $head(\sigma)$). An active integrity constraint specifies both an integrity constraint (in the body) and the actions to be performed (in the head) if the integrity constraint is violated. We use $St(\sigma)$ to denote the standard integrity constraint of the form (1), derived from σ by removing all the head update atoms. Also, $ground(\sigma)$ denotes the set of all ground instances of σ, that is, all AICs that can be derived from σ by replacing every variable with a constant.

For a set of active integrity constraints Σ, $St(\Sigma)$ denotes the corresponding set of integrity constraints, that is $St(\Sigma) = \{St(\sigma) \mid \sigma \in \Sigma\}$. Also, $ground(\Sigma)$ is the set of all ground instances of constraints in Σ, i.e., $ground(\Sigma) = \bigcup_{\sigma \in \Sigma} ground(\sigma)$.

Semantics. Before introducing the formal semantics of AICs, we introduce some auxiliary notions.

Definition 2.2. Given a database D and a coherent set of update atoms U, the truth value of:

- a positive ground literal $a(\overline{c})$ is *true* w.r.t. (D, U) if $a(\overline{c}) \in U(D)$, and is *false* otherwise;
- a negative ground literal *not* $a(\overline{c})$ is *true* w.r.t. (D, U) if $a(\overline{c}) \notin U(D)$, and is *false* otherwise;
- a ground update atom $\pm a(\overline{c})$ is *true* w.r.t. (D, U) if $\pm a(\overline{c}) \in U$, and is *false* otherwise;
- comparison atoms, conjunctions, and disjunctions of literals is defined in the obvious way,
- a ground AIC σ is *true* w.r.t. (D, U) if some literal in $body(\sigma)$ is *false* w.r.t. (D, U), and is *false* otherwise. □

For any database D and set of AICs Σ, a set R of ground update atoms is a *repairing update set* for $\langle D, \Sigma\rangle$ if it is a repairing update set for $\langle D, St(\Sigma)\rangle$. The set of repairing update sets for $\langle D, \Sigma\rangle$ is denoted as $\mathbf{R}(D, \Sigma)$.

Definition 2.3. Let D be a database, Σ a set of AICs, and R a repairing update set for $\langle D, \Sigma\rangle$.

- A ground update atom $\pm a(\overline{c}) \in R$ is *founded* iff there exists $\sigma \in ground(\Sigma)$ s.t. $\pm a(\overline{c})$ occurs in $head(\sigma)$ and $body(\sigma)$ is *true* w.r.t. $(D, R \setminus \{\pm a(\overline{c})\})$. We also say that $\pm a(\overline{c})$ is *supported* by σ w.r.t. R.
- R is *founded* iff all its atoms are *founded*.
- R is *unfounded* iff it is not founded. □

Given a database D and a set of AICs Σ, the set of founded repairing update sets for $\langle D, \Sigma\rangle$ is denoted as $\mathbf{FR}(D, \Sigma)$. Clearly, the set of founded repairing update sets is contained in the set of repairing update sets, that is, $\mathbf{FR}(D, \Sigma) \subseteq \mathbf{R}(D, \Sigma)$.

Founded certain and possible answers are defined analougosly to the case of standard constraints by considering founded repairs.

3 PRIORITIZED ACTIVE INTEGRITY CONSTRAINTS

In this section, we present an extension of integrity constraints which allows us to specify, for each constraint, the update actions to be performed to satisfy it and preferences between them.

3.1 Syntax

The following definition introduces prioritized active integrity constraints, which extend AICs by allowing to express preferences among update atoms.

Definition 3.1. A *prioritized active integrity constraint* (PAIC) is of the form:

$$\forall \overline{X} \left[\varphi(\overline{X}_0) \wedge \bigwedge_{i=1}^{m} b_i(\overline{X}_i) \ \wedge \bigwedge_{i=m+1}^{n} not\ b_i(\overline{X}_i) \Rightarrow \bigotimes_{i=n+1}^{p} \pm a_i(\overline{X}_i) \right] \quad (2)$$

where $\otimes$ is either $\vee$ or $\succ$, $\overline{X} = \bigcup_{i=1}^{m} \overline{X}_i$, $\overline{X}_i \subseteq \overline{X}$ for every $0 \leq i \leq p$, $\varphi(\overline{X}_0)$ is a conjunction of comparison atoms, the $b_i(\overline{X}_i)$'s are atoms, and the $\pm a_i(\overline{X}_i)$'s are update atoms. □

W.l.o.g. we assume that the operators in the head are all of the same type. Thus, the head of the PAIC is a disjunction of update atoms (if "$\otimes$" = " $\vee$ ") or a linearly ordered disjunction of atoms (if "$\otimes$" = " $\succ$ ") expressing our preferences over the alternative updates. Intuitively, $\pm a_{n+1}(\overline{X}_{n+1}) \succ \cdots \succ \pm a_p(\overline{X}_p)$ means that the action $\pm a_i(\overline{X}_i)$, is preferable to the actions $\pm a_{i+1}(\overline{X}_{i+1}), \ldots, \pm a_p(\overline{X}_p)$ for $i \in [n+1, p-1]$.

As usual, given a PAIC σ, the conjunction (resp., disjunction) on the left-hand (resp., right-hand) side of $\Rightarrow$ is called the *body* (resp., *head*) of σ and is denoted $body(\sigma)$ (resp., $head(\sigma)$). In the following, we will omit the universal quantification in front of integrity constraints and assume that all variables are universally quantified.

Definition 3.2. Let σ be a PAIC and Σ be a set of PAICs. Then,

- $Ac(\sigma)$ denotes the active integrity constraint derived from σ by replacing symbol $\succ$ with $\vee$, while $Ac(\Sigma) = \{Ac(\sigma) \mid \in \Sigma\}$.
- $St(\sigma)$ denotes the standard constraint derived from σ by deleting the update atoms appearing in the head. Moreover, $St(\Sigma) = \{St(\sigma) \mid \sigma \in \Sigma\}$. □

Definition 3.3. A (prioritized) active integrity constraint is said to be in *canonical* form if for each update literal $\pm a(\overline{X})$ appearing in the head, a literal $cLit(\pm a(\overline{X}))$ also appears in the body. A set of (prioritized) active integrity constraints is said to be *canonical* if all constraints are in canonical form. □

In the rest of the paper, prioritized active integrity constraints are assumed to be in canonical form. The motivation for restricting our attention to canonical PAICs is that [18] showed that for every ground AIC σ, every head update atom $\pm a(\overline{y})$ such that $cLit(\pm a(\overline{y})) \notin body(\sigma)$ is useless and can be deleted.

3.2 Semantics

In the following, we first provide some auxiliary definitions and then report the formal definition of *preferred repairs*.

Given an integrity constraint (resp., AIC, PAIC) σ, we use $ground(\sigma)$ to denote the set of all integrity constraints (resp., AICs, PAICs) that can be obtained from σ by replacing every variable with a constant. Given a set Σ of integrity constraints (resp., AICs, PAICs), we define $ground(\Sigma) = \bigcup_{\sigma \in \Sigma} ground(\sigma)$.

The set of all (resp., founded) repairing update sets for a database D and a set Σ of PAICs, denoted $\mathbf{R}(D, \Sigma)$ (resp., $\mathbf{FR}(D, \Sigma)$), is the set of all (resp., founded) repairing update sets for $\langle D, St(\Sigma)\rangle$ (resp.,$\langle D, Ac(\Sigma)\rangle$).

Definition 3.4 (Preferences between repairing update sets). Let D be a database and Σ be a set of PAICs. The binary relation $\sqsupseteq$ among repairing update sets is defined as follows. For any repairing update sets $R_1, R_2, R_3 \in \mathbf{R}(D, \Sigma)$:

(1) $R_1 \sqsupseteq R_1$.
(2) $R_1 \sqsupseteq R_2$ if:
  (a) $R_1 \in \mathbf{FR}(D, \Sigma)$ and $R_2 \notin \mathbf{FR}(D, \Sigma)$, or
  (b)(i) $R_1, R_2 \in \mathbf{FR}(D, \Sigma)$ or $R_1, R_2 \notin \mathbf{FR}(D, \Sigma)$ and
    (ii) there are two ground update atoms $\pm a(\overline{c}) \in R_1$ and $\pm b(\overline{u}) \in R_2$ and a (ground) prioritized active integrity constraint σ such that
    i) $head(\sigma) = \ldots \pm a(\overline{c}) \ldots \succ \ldots \pm b(\overline{u}) \ldots$ and
    ii) σ supports $\pm a(\overline{c})$ w.r.t. R_1 and $\pm b(\overline{u})$ w.r.t. R_2.
(3) If $R_1 \sqsupseteq R_2$ and $R_2 \sqsupseteq R_3$, then $R_1 \sqsupseteq R_3$.

We write $R_1 \sqsupset R_2$ if $R_1 \sqsupseteq R_2$ and $R_2 \not\sqsupseteq R_1$. A repairing update set R is a *preferred* repairing update set if there is no repairing update set R' such that $R' \sqsupset R$. □

Intuitively, the preference relation $\sqsupseteq$ among repairing update sets is defined by first establisching the direct preference relation between repairing update sets and next, by applying the transitive property. Any founded repairing update set is preferable to an unfounded one. The direct comparison of two founded (resp. unfounded) repairing update sets R_1 and R_2 is possible whenever they contain comparable update actions, i.e $\pm a(\overline{c}) \in R_1$ and $\pm b(\overline{u}) \in R_2$ are supported by a ground PAIC σ w.r.t. R_1 and R_2, respectively, and σ specifies the priority level between $\pm a(\overline{c})$ and $\pm b(\overline{u})$.

The set of preferred repairing update sets for a database D and a set of PAICs Σ is denoted by $\mathbf{PR}(D, \Sigma)$. Repairs obtained from preferred repairing update sets are called *preferred repairs*.

Example 3.5. Consider the following set of PAICs Σ:

$$\begin{array}{lcl} \mathtt{not\ a} \wedge \mathtt{not\ b} \wedge \mathtt{c} & \Rightarrow & \mathtt{+a} \succ \mathtt{+b} \\ \mathtt{not\ c} \wedge \mathtt{d} & \Rightarrow & \mathtt{+c} \succ \mathtt{-d} \end{array}$$

where both PAICs are in canonical form. The following (active) integrity constraints can be derived:

- $Ac(\Sigma)$ consists of the active integrity constraints

$$\begin{array}{lcl} \mathtt{not\ a} \wedge \mathtt{not\ b} \wedge \mathtt{c} & \Rightarrow & \mathtt{+a} \vee \mathtt{+b} \\ \mathtt{not\ c} \wedge \mathtt{d} & \Rightarrow & \mathtt{+c} \vee \mathtt{-d} \end{array}$$

- $St(\Sigma)$ consists of the integrity constraints

$$\begin{array}{ll} \mathtt{not\ a} \wedge \mathtt{not\ b} \wedge \mathtt{c} & \Rightarrow \\ \mathtt{not\ c} \wedge \mathtt{d} & \Rightarrow \end{array}$$

Considering the database $D = \{\mathtt{c}, \mathtt{d}\}$, we have $\mathbf{R}(D, \Sigma) = \{R_1, R_2, R_3\}$ where $R_1 = \{-\mathtt{c}, -\mathtt{d}\}, R_2 = \{+\mathtt{a}\}, R_3 = \{+\mathtt{b}\}$. Consequenty, we have three repairs $D_1 = R_1(D) = \{\ \}$, $D_2 = R_2(D) = \{\mathtt{c}, \mathtt{d}, \mathtt{a}\}$ and $D_3 = R_3(D) = \{\mathtt{c}, \mathtt{d}, \mathtt{b}\}$. Moreover, $\mathbf{FR}(D, \Sigma) = \{R_2, R_3\}$ (i.e., repair R_1 is not founded as update $-\mathtt{c}$ is not supported) and, since $R_2 \sqsupset R_3$, $\mathbf{PR}(D, \Sigma) = \{R_2\}$. □

Example 3.6. Consider the following set of PAICs Σ:

$$\mathtt{mgr(E,P)} \wedge \mathtt{prj(P,D)} \wedge \mathtt{not\ emp(E,D)} \Rightarrow \mathtt{+emp(E,D)} \succ \mathtt{-mgr(E,P)}$$
$$\mathtt{emp(E,D_1)} \ \wedge \ \mathtt{emp(E,D_2)} \ \wedge \ \mathtt{D_1} \neq \mathtt{D_2} \Rightarrow \mathtt{-emp(E,D_1)} \vee \mathtt{-emp(E,D_2)}$$

The first constraint states that every manager E of a project P carried out by a department D must be an employee of D, whereas the second one says that every employee must be in only one department.

Consider the database $D = \{\mathtt{mgr}(\mathtt{e}_1, \mathtt{p}_1),\ \mathtt{prj}(\mathtt{p}_1, \mathtt{d}_1),\ \mathtt{emp}(\mathtt{e}_1, \mathtt{d}_2)\}$. There are three repairing update sets for (D, Σ): $R_1 = \{-\mathtt{mgr}(\mathtt{e}_1, \mathtt{p}_1)\}$, $R_2 = \{-\mathtt{prj}(\mathtt{p}_1, \mathtt{d}_1)\}$, and $R_3 = \{+\mathtt{emp}(\mathtt{e}_1, \mathtt{d}_1),\ -\mathtt{emp}(\mathtt{e}_1, \mathtt{d}_2)\}$. The only founded repairing update sets are R_1 and R_3, as only the update atoms $-\mathtt{mgr}(\mathtt{e}_1, \mathtt{p}_1)$, $+\mathtt{emp}(\mathtt{e}_1, \mathtt{d}_1)$, and $-\mathtt{emp}(\mathtt{e}_1, \mathtt{d}_2)$ are supported. Observe that $-\mathtt{prj}(\mathtt{p}_1, \mathtt{d}_1)$ is not allowed in Σ. Moreover, since R_3 is preferable to R_1, then R_3 is the only preferred repairing update set. □

As for standard and active integrity constraints, preferred certain and possible answers are defined by only considering preferred repairs.

4 COMPUTING REPAIRING UPDATE SETS

In this section, we show how preferred repairing update sets can be computed using prioritized logic programs under preferred stable model semantics.

We first recall how (founded) repairing update sets are computed using logic programs under the stable model semantics. One important advantage of this approach is that repairing update sets can be computed by using existing systems for evaluating logic programs (e.g., [50]).

4.1 Computing Founded Repairing Update Sets

The computation of repairing update sets for a set of standard integrity constraints Σ using a set logic rules $DP(\Sigma)$ under answer set semantics has been proposed in [44]. The next definition shows how $DP(\Sigma)$ is defined.

Definition 4.1. Given a set of integrity constraints Σ, then $DP(\Sigma)$ is the program derived from Σ by replacing each integrity constraint in Σ of the form (1) with a rule of the form:

$$\bigvee_{j=1}^{m} -b_j(\overline{X}_j) \vee \bigvee_{j=m+1}^{n} +b_j(\overline{X}_j) \leftarrow \bigwedge_{j=1}^{m} B_j(\overline{X}_j) \wedge \bigwedge_{j=m+1}^{n} not\ B_j(\overline{X}_j) \wedge \varphi(\overline{X}_0)$$

where, for each predicate b, B is a new predicate symbol univokely associated with b and by adding the rules:

$$B(\overline{X}) \leftarrow (b(\overline{X}) \wedge not - b(\overline{X})) \vee +b(\overline{X}),$$
$$\leftarrow -b(\overline{X}) \wedge +b(\overline{X}). \qquad \square$$

In the previous definition the rule with empty head is a denial rule which is satisfied only if the body is false, whereas the predicate B has been introduced just to simplify the mapping. Observe also that the body of the rule defining predicate B contains a disjunction operator. This is just a shorthand to make programs more compact as predicate B could be be equivalently defined using two distinct rules without disjunction in the body.

We point out that the rewriting introduced in Definition 4.1 is slightly different, but equivalent to the one presented in [44]. Here we have introduced additional predicates just to make rules easy to understand and to simplify the rewritings we are going to present next. It is worth noting that for atoms of the form $+a(t)$, $-a(t)$ and $a(t)$, the symbols $+a$, $-a$ and a are assumed to be different predicates.

For any interpretation M, we denote as $UpdateAtoms(M)$ the set of update atoms in M. Given a set S of interpretations, we define $UpdateAtoms(S) = \{UpdateAtoms(M) \mid M \in S\}$.

Theorem 4.2. *[44] Given a database D and a set of integrity constraints Σ, then:*

- *(Soundness) for every stable model M of $DP(\Sigma) \cup D$, $UpdateAtoms(M)$ is a repairing update set for $\langle D, \Sigma\rangle$;*
- *(Completeness) for every repairing update set S for $\langle D, \Sigma\rangle$ there exists a stable model M of $DP(\Sigma) \cup D$ such that $S = UpdateAtoms(M)$.* □

The restriction of repairing update sets to founded repairing update sets is performed by adding additional rules do $DP(\Sigma)$ and an additional set of rules denoted by $FP(\Sigma)$.

Definition 4.3. Given a set of active integrity constraints Σ, $FP(\Sigma)$ is the set of logic rules obtained from Σ as follows.

- For each rule $\sigma \in \Sigma$ of the form (1) and for each update atom $\pm a_j(\overline{X}_j)$ in the head of σ, $FP(\Sigma)$ has a rule of the following form:

$$a_{jF}^{\pm}(\overline{X}_j) \leftarrow cLit(\pm a_j(\overline{X}_j)) \wedge \varphi(\overline{X}_0) \wedge$$
$$\bigwedge \{b_i(\overline{X}_i) \mid i \in [1, m] \wedge b_i(\overline{X}_i) \neq cLit(\pm a_j(\overline{X}_j))\} \wedge$$
$$\bigwedge \{not\ b_i(\overline{X}_i) \mid i \in [m+1, n] \wedge not\ b_i(\overline{X}_i) \neq cLit(\pm a_j(\overline{X}_j))\}.$$

- For each update atom $\pm a(\overline{X})$ defined in $DP(\Sigma)$, the following rule belongs to $FP(\Sigma)$:

$$unfounded \leftarrow \pm a(\overline{X}) \ \wedge \ not\ a_F^{\pm}(\overline{X})$$

- The following rule belongs to $FP(\Sigma)$:

$$founded \leftarrow not\ unfounded$$

- No other rule belongs to $FP(\Sigma)$.

Given a set of AICs Σ, then $FDP(\Sigma) = DP(St(\Sigma)) \cup FP(\Sigma)$, whereas $FDP'(\Sigma)$ is obtained by adding the denial rule $\leftarrow$ *unfounded* to $FDP(\Sigma)$. □

In the previous definition the subscript *"F"* is used to highlight the fact that $a_F^{\pm}$-atoms are founded. The set of rules $FP(\Sigma)$ is used to derive founded update atoms. Rules in $FP(\Sigma)$ state that an update atom is founded if it does not support itself.

Observe that given a set of PAICs Σ and a database D, then the stable models of $DP(\Sigma) \cup D$ coincide with all the possible repairing update sets for $\langle D, \Sigma\rangle$, whereas the stable models of $FDP'(\Sigma) \cup D$ coincide with the founded repairing update sets as the denial rule $\leftarrow$ *unfounded* in $FDP'(\Sigma)$ discards every stable model of $FDP(\Sigma) \cup D$ that does not correspond to a founded repairing update set.

The next theorem (proved in [19]) states the correctness of the technique proposed above.

Theorem 4.4. *[19] Given a database D and a set of PAICs Σ, then:*

- *(Soundness) for every stable model M of $FDP'(\Sigma) \cup D$, a set $UpdateAtoms(M)$ is a founded repairing update set for $\langle D, \Sigma\rangle$;*

- *(Completeness) for every founded repairing update set S for $\langle D, \Sigma\rangle$ there exists a stable model M of $FDP'(\Sigma) \cup D$ such that $S = UpdateAtoms(M)$.* □

4.2 Computing Preferred Repairing Update Sets

We are now ready to show how preferred repairs can be computed using prioritized logic programs under preferred stable model semantics. In what follows, when defining a prioritized logic program (P, Φ), if A_1 and A_2 are atoms and B is a conjunction of literals, when we say that $A_1 \succeq A_2 \leftarrow B$ belongs to Φ, which intuitively means that $A_1 \succeq A_2$ holds under a condition B, we mean that the rules $A_1' \leftarrow B \wedge A_1$ and $A_2' \leftarrow B \wedge A_2$ belong to P, and the priority $A_1' \succeq A_2'$ belongs to Φ.

Definition 4.5. Given a set of prioritized active integrity constraints Σ, $PLP(\Sigma)$ is the prioritized logic program $\langle FDP(\Sigma), \Phi(\Sigma)\rangle$, where $\Phi(\Sigma)$ contains the following priorities:

(1) *founded* $\succ$ *unfounded*,

(2) for each PAIC $\sigma \in \Sigma$ of the form (2) and for each pair of update atoms $\pm a_j(\overline{X}_j), \pm a_k(\overline{X}_k) \in head(\sigma)$ such that $\pm a_j(\overline{X}_j) \succ \pm a_k(\overline{X}_k)$, two rules of the following form are added to $\Phi(\Sigma)$:

$$\begin{aligned}\pm a_j(\overline{X}_j) \succ \pm a_k(\overline{X}_k) \leftarrow\ & cLit(\pm a_j(\overline{X}_j)) \wedge\ cLit(\pm a_k(\overline{X}_k)) \wedge \\ & \bigwedge \{\, B_i(\overline{X}_i) \mid i \in [1, m] \\ & \quad \wedge\, b_i(\overline{X}_i) \notin cLit(\{\pm a_j(\overline{X}_j), \pm a_k(\overline{X}_k)\}) \,\} \wedge \\ & \bigwedge \{\, not\ B_i(\overline{X}_i) \mid i \in [m+1, n] \\ & \quad \wedge\, not\ b_i(\overline{X}_i) \notin cLit(\{\pm a_j(\overline{X}_j), \pm a_k(\overline{X}_k)\}) \,\} \wedge \\ & \varphi(X_0) \wedge RepairType,\end{aligned}$$

where $RepairType \in \{founded, unfounded\}$. That is, the two rules are obtained by replacing *RepairType* with *founded* and *unfounded*, respectively. □

Example 4.6. Let us consider the database D and the PAICs Σ presented in Example 3.5. We derive from Σ the prioritized logic program $\langle FDP(\Sigma), \Phi(\Sigma)\rangle$ where the set of prioritized rules $\Phi(\Sigma)$ is as follows:

$$\begin{array}{lcl} +a \geq +b & \leftarrow & not\ a \wedge not\ b \wedge C \wedge founded \\ +a \geq +b & \leftarrow & not\ a \wedge not\ b \wedge C \wedge unfounded \\ +c \geq -d & \leftarrow & not\ c \wedge d \wedge founded \\ +c \geq -d & \leftarrow & not\ c \wedge d \wedge unfounded \\ founded & \succ & unfounded \end{array}$$

The stable models of $FDP(\Sigma) \cup D$ are:

- $M_1 = \{c,\ d,\ -c,\ -d,\ d_F^-,\ unfounded\}$,
- $M_2 = \{c,\ d,\ +a,\ C,\ D,\ A,\ a_F^+,\ founded\}$ and
- $M_3 = \{c,\ d,\ +b,\ C,\ D,\ B,\ b_F^+,\ founded\}$.

Moreover, having $M_2 \sqsupset M_3$, $M_2 \sqsupset M_1$ and $M_3 \sqsupset M_1$, the preferred stable model is M_2. We observe that while M_2 and M_3 are both preferred to M_1 because of the priority *founded* $\succ$ *unfounded*, M_2 is preferred to M_3 because of the priority $+a \geq +b \leftarrow not\ a \wedge not\ b \wedge C \wedge$ *founded*. □

Our main result follows.

Theorem 4.7. Given a database D and a set of prioritized active integrity constraints Σ, then:

- (Soundness) for every preferred stable model M of $\langle FDP(\Sigma) \cup D, \Phi(\Sigma)\rangle$, $UpdateAtoms(M)$ is a preferred repairing update set for $\langle D, \Sigma\rangle$;
- (Completeness) for every preferred repairing update set S for $\langle D, \Sigma\rangle$ there exists a preferred stable model M of $\langle FDP(\Sigma) \cup D, \Phi(\Sigma)\rangle$ such that $S = UpdateAtoms(M)$. □

5 PRIORITIZED ACTIVE FUNCTIONAL DEPENDENCIES (PAFDS)

So far, we have considered general PAICs. We now study the special case where PAICs defines (conditional) functional dependencies with (prioritized) actions to be performed if the functional dependency is violated and some conditions hold. First, we introduce the syntax of functional dependencies, which are a restricted class of integrity constraints that allow us to express many constraints commonly arising in practice, such as key constraints.

A *functional dependency* is an integrity constraint of the form:

$$\forall \overline{X}, \overline{Y}_1, \overline{Y}_2, \overline{Z}_1, \overline{Z}_2 [p(\overline{X}, \overline{Y}_1, \overline{Z}_1) \wedge p(\overline{X}, \overline{Y}_2, \overline{Z}_2) \wedge \overline{Y}_1 \neq \overline{Y}_2 \Rightarrow]$$

where p is a predicate and $\overline{X}, \overline{Y}_1, \overline{Y}_2, \overline{Z}_1, \overline{Z}_2$ are lists of variables with $\overline{Y}_1$ and $\overline{Y}_2$ (resp. $\overline{Z}_1$ and $\overline{Z}_2$) being of the same length. Equivalently, it can also be written as:

$$\forall \overline{X}, \overline{Y}_1, \overline{Y}_2, \overline{Z}_1, \overline{Z}_2\ [p(\overline{X}, \overline{Y}_1, \overline{Z}_1) \wedge p(\overline{X}, \overline{Y}_2, \overline{Z}_2) \Rightarrow \overline{Y}_1 = \overline{Y}_2]$$

Prioritized active functional dependencies are defined as follows.

Definition 5.1. A *Prioritized Active Functional Dependency* (PAFD) is an PAIC of the form:

$$\begin{aligned} p(\overline{X}, \overline{Y}_1, \overline{Z}_1) \wedge p(\overline{X}, \overline{Y}_2, \overline{Z}_2) \wedge \overline{Y}_1 \neq \overline{Y}_2 \wedge \varphi(\overline{X}, \overline{Y}_1, \overline{Y}_2, \overline{Z}_1, \overline{Z}_2) \Rightarrow \\ -p(\overline{X}, \overline{Y}_1, \overline{Z}_1)\ \star\ -p(\overline{X}, \overline{Y}_2, \overline{Z}_2) \end{aligned}$$

where $\varphi(\overline{X}, \overline{Y}_1, \overline{Y}_2, \overline{Z}_1, \overline{Z}_2)$ is a conjunction of comparison atoms and $\star \in \{\vee, \succ\}$. □

In previous definition, the conjunction $\varphi(\overline{X}, \overline{Y}_1, \overline{Y}_2, \overline{Z}_1, \overline{Z}_2)$ defines a condition over the active functional dependency implying $\overline{Y}_1 \neq \overline{Y}_2$. When the (conditional) functional dependency is violated, the PAFD specify the preferred action to be performed to restore consistency. In the following, the conjunction φ occurring in a PAFD r will be denoted by $\varphi(r)$.

Example 5.2. Consider a relation $emp(id, name, salary)$, storing information regarding employees, and the following PAFD:

$$\begin{aligned} emp(I, N_1, S_1) \wedge emp(I, N_2, S_2) \wedge N_1 \neq N_2 \wedge S_1 > S_2 \Rightarrow \\ -emp(I, N_1, S_1) \succ -emp(I, N_2, S_2). \end{aligned}$$

The intuitive meaning of this PAFD is that the relation *emp* cannot contain two tuples with the same *id* and different values for the attribute *name*. When this happens, it is preferable to delete the tuple with the higher *salary* in order to restore consistency. □

The following definition introduces a notion of "coherence" for a set of PAFDs.

Definition 5.3. Given a database D and a set of PAFDs FD, we say that FD is *coherent* with respect to D if for each $s = body_s \Rightarrow -A \succ$

$-B \in ground(FD)$ such that $D \models body_s$ the following conditions hold:

(1) there is no $r = body_r \Rightarrow -B \succ -A \in ground(FD)$ such that $D \models body_r$; and
(2) if there is $r = body_r \Rightarrow -B \succ -C \in ground(FD)$ such that $D \models body_r$, then there is $t = body_t \Rightarrow -A \succ -C \in ground(FD)$ such that $D \models body_t$; □

The intuitive meaning of the first item in the previous definition is that if there is a ground PAFD stating that the deletion of an atom A is preferable to the deletion of an atom B, then there cannot be a ground PAFD stating that the deletion of B is preferable to the deletion of an atom A. The intuitive meaning of the second item is that if there is a ground PAFD stating that the deletion of an atom A is preferable to the deletion of an atom B, and there is a PAFD asserting that the deletion of B is preferable to the deletion of another atom C, then the deletion of A has to be supported by a third PAFD asserting that it is preferable over the deletion of an atom C. In this way, both A and B have to be removed and the uncertainty is resolved.

Observe that Definition 5.3 implies that $ground(FD)$ cannot contain a ground PAFD of the form $body \Rightarrow -A \succ -A$ such that $D \models body$.

Theorem 5.4. *Let FD be a set of PAFDs and D a database. Whether FD is coherent with respect to D can be checked in polynomial time.*

We are now ready to present the main result of this section.

Theorem 5.5. *Let D be a database and FD a set of PAFDs coherent with respect to D. Then,*

(1) *(D, FD) admits a unique preferred repair D',*
(2) *D' can be computed in polynomial time with respect to the size of D.*

Proof.

(1) Let $U = \{-A \mid (body \Rightarrow -A \succ -B) \in ground(FD) \ \wedge \ D \models body\}$.
Let us prove that U is the unique founded repairing update set for $\langle D, FD\rangle$. First of all, the updated database $D' = U(D)$ is consistent by construction. Indeed, for each ground PAFD $\sigma \in ground(FD)$ of the form $(body \Rightarrow -A \succ -B)$ which is violated by D, the updated action $-A$ is in U. Therefore, D' satisfies σ.
Now we prove that U is founded that is each element $-A \in U$ is founded.
Let $\overline{V} = U \ \cup \ \{-B \mid (body \Rightarrow -A \succ -B) \in ground(FD) \ \wedge \ D \models body\}$. Let us consider the relation $\theta = \{(-A, -B) \mid (body \Rightarrow -A \succ -B) \in ground(FD) \ \wedge \ D \models body\} \subseteq \overline{V} \times \overline{V}$ (as usual we denote $(-A, -B) \in \theta$ as $-A \ \theta \ -B$).
As FD is coherent w.r.t. D, the relation θ is a strict partial order (see Definition 5.3).
For each $-A \in U$, by construction, there must be another element $-B \in \overline{V}$ s.t. $-A \ \theta \ -B$. As θ is a strict partial order, there must be a maximal chain $C \subseteq \overline{V}$ s.t. $-A, -B \in C$. As $\overline{V}$ is finite, this chain is finite too and admits a minimum element $-C$. Then, $-A \ \theta - C$ and there is no element $-D \in \overline{V}$ s.t. $-D \ \theta - C$. In other words, there is a ground constraint r of the form $(body_r \Rightarrow -A \succ -C) \in ground(FD)$ s.t. $D \models body_r$ and there is no ground constraint s of the form $(body_s \Rightarrow -C \succ -D) \in ground(FD)$ s.t. $D \models body_s$. This means that $-C \notin U$. Therefore, r supports $-A$ and $-A$ is founded. As the deletion of each $-A \in U$ leads to the violation of the ground constraint supporting it, U is trivially minimal. Moreover, as U is the unique founded repairing update set, it is also preferred.
(2) The process to derive U and so the repair $D' = U(D)$ can be trivially performed in polynomial time in the size of D. □

Corollary 5.6. *Let D be a database, FD a set of PAFDs coherent with respect to D, and Q a query. Then, the preferred consistent query answers to Q on D w.r.t. FD can be computed in polynomial time.*

Proof. It follows from the fact that since there is a unique preferred repair D' computable in polynomial time (Theorem 5.5), the answer to Q over D', can be computed in polynomial time as well. □

6 RELATED WORK

The notion of automatic consistency maintenance in the presence of integrity constraints has been extensively considered in the context of database management systems. Many approaches proposed in the literature make use of ECA (event-condition-action) rules for checking and enforcing integrity constraints. In addition, current DBMS languages offer the possibility of defining triggers (special ECA rules) well suited to respond automatically, performing actions, to events that are taking place inside (or even outside) the database. The effort of "adding" active rules into conventional database systems has raised considerable interest both in the scientific community and in the commercial world. As a consequence, a number of prototypes and systems have been developed [64]. However, the problem with active rules is the difficulty to understand the behaviour when a significant number of trigger act simultaneously [23, 57, 60].

In [25] and [24] a framework for database maintenance enforcing constraints by issuing actions to be performed to correct violation has been proposed, while the problem of maintaining integrity constraints in database systems has been considered in [58].

The application of the ECA paradigm of active databases to policies—collection of general principles specifying the desired behavior of systems—has been investigated in [27], whereas the problem of specifying policies for database maintenance using situation calculus has been studied in [6].

The problem of providing an homogeneous framework for integrating, in a database environment, active rules and deductive rules has been addressed in [59], whereas deterministic and nondeterministic semantics for active rules have been investigated in [39].

A declarative framework for updating views over indefinite databases has been proposed in [20]. This work introduces the concepts of *constrained repairing update sets*. Constrained repairing update sets fulfill the view-update request, changing the database minimally and avoiding arbitrary commitments. A similar approach

has been presented for abductive logic programming extended with integrity constraints in [21]. This work introduces the concept of *constrained explanation* for an observation as an explanation having no arbitrariness.

Querying inconsistent data and knowledge bases has been deeply investigated in the areas of databases and artificial intelligence [1, 63].

The notions of *repair* and *consistent query answer* have been introduced in [1]. The same work has also proposed a method, based on query rewriting, to compute consistent query answers. The proposed technique is simple, but has a limited applicability (first-order queries without disjunction or quantification, and binary universal integrity constraints). That approach has been generalized by Fuxman and Miller [41, 42] to allow restricted existential quantification in queries in the presence of primary key FDs.

Several authors have considered the use of logic programs to capture repairs as answer sets of logic programs with negation and disjunction [2, 44, 55]. These approaches are quite general, being able to handle arbitrary universal constraints and first-order queries. However, such approaches do not deal with preferences. Moreover, although we focus on (RA) queries encoded via logic programs, our approach allows to compute the preferred consistent query answers for all queries for which the number and size of stable models is finite, e.g. see [3, 10–12, 14–16, 32, 34, 56].

The use of preferences and priorities, as well as the definition of constructs to define infeasible repairs and answers, have been proposed by [17, 36, 44, 51–54]. In most of the proposed approaches, the use of preferences further increases the computational complexity.

Three-valued interpretations of database theories have been proposed as well and techniques for query answering under LCWA (local closed-world assumption) and computing a three-valued interpretation that approximates all two-valued interpretations of a database theory has been presented in [28, 40, 45]. For surveys on repairing and querying inconsistent databases we refer to [4, 5, 26].

Approaches for repairing inconsistent databases by means of tuple updates have been proposed in [37, 38, 46, 47, 49].

Active integrity constraints have been introduced in [18] and fourthly investigated in [19]. There is some additional work on AICs by other authors that are relevant and that might be worth mentioning: algorithms for computing all classes of repairs discussed in [22] were given in [31]; and a framework for applying AICs to more knowledge representation systems other than databases was proposed in [7], generalizing also the approach in [61] (see also [35]). The computation of repairs is also related to the computation of grounded fixpoints that have been shown to be the "natural" semantics in many knowledge representation formalisms [8]. Finally, [29] proposed a notion of stratification of AICs (also discussed in [30]) that bears resemblance to the notion of an AIC firing or blocking another.

7 CONCLUSIONS

In this paper, we have made new contributions in the context of Prioritized Active Integrity Constraints. Specifically, we proposed an approach to compute preferred repairing update sets via the computation of preferred stable models of prioritized logic programs (which are derived from PAICs).

Furthermore, we have proposed a relevant class of PAICs, called prioritized active functional dependencies, which allows us to express functional dependencies, admits a unique preferred repairing update set, and for which consistent answers can be computed in polynomial time. Further researches will investigate the use of more general query languages [13, 43, 48, 50].

REFERENCES

[1] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. 1999. Consistent Query Answers in Inconsistent Databases. In *PODS*. 68–79.

[2] Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. 2003. Answer sets for consistent query answering in inconsistent databases. *Theory and Practice of Logic Programming* 3, 4-5 (2003), 393–424.

[3] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat. 2011. On rules with existential variables: Walking the decidability line. *Artif. Intell.* 175, 9-10 (2011), 1620–1654.

[4] Leopoldo E. Bertossi. 2006. Consistent query answering in databases. *SIGMOD Record* 35, 2 (2006), 68–76.

[5] Leopoldo E. Bertossi. 2011. *Database Repairing and Consistent Query Answering*. Morgan & Claypool Publishers.

[6] Leopoldo E. Bertossi and Javier Pinto. 1999. Specifying Active Rules for Database Maintenance. In *Proc. of the Int. Ws. on Foundations of Models and Languages for Data and Objects (FMLDO)*. 65–81.

[7] Bart Bogaerts and Luís Cruz-Filipe. 2018. Fixpoint semantics for active integrity constraints. *Artif. Intell.* 255 (2018), 43–70.

[8] Bart Bogaerts, Joost Vennekens, and Marc Denecker. 2015. Grounded fixpoints and their applications in knowledge representation. *Artif. Intell.* 224 (2015), 51–71.

[9] Gerhard Brewka and Thomas Eiter. 1999. Preferred Answer Sets for Extended Logic Programs. *Artificial Intelligence* 109, 1-2 (1999), 297–356.

[10] Marco Calautti, Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. 2014. Checking Termination of Logic Programs with Function Symbols through Linear Constraints. In *RuleML*, Vol. 8620. 97–111.

[11] Marco Calautti, Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. 2015. Logic Program Termination Analysis Using Atom Sizes. In *IJCAI*. AAAI Press, 2833–2839.

[12] Marco Calautti, Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. 2016. Using linear constraints for logic program termination analysis. *Theory Pract. Log. Program.* 16, 3 (2016), 353–377.

[13] Marco Calautti, Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. 2020. Preference-based Inconsistency-Tolerant Query Answering under Existential Rules. In *KR*. to appear.

[14] Marco Calautti, Sergio Greco, Francesca Spezzano, and Irina Trubitsyna. 2015. Checking termination of bottom-up evaluation of logic programs with function symbols. *Theory Pract. Log. Program.* 15, 6 (2015), 854–889.

[15] Marco Calautti, Sergio Greco, and Irina Trubitsyna. 2013. Detecting decidable classes of finitely ground logic programs with function symbols. In *15th International Symposium on Principles and Practice of Declarative Programming, PPDP '13, Madrid, Spain, September 16-18, 2013*, Ricardo Peña and Tom Schrijvers (Eds.). ACM, 239–250.

[16] Marco Calautti, Sergio Greco, and Irina Trubitsyna. 2017. Detecting Decidable Classes of Finitely Ground Logic Programs with Function Symbols. *ACM Trans. Comput. Log.* 18, 4 (2017), 28:1–28:42.

[17] Luciano Caroprese, Sergio Greco, and Cristian Molinaro. 2007. Prioritized Active Integrity Constraints for Database Maintenance. In *DASFAA*. 459–471.

[18] Luciano Caroprese, Sergio Greco, Cristina Sirangelo, and Ester Zumpano. 2006. Declarative Semantics of Production Rules for Integrity Maintenance. In *ICLP*. 26–40.

[19] Luciano Caroprese, Sergio Greco, and Ester Zumpano. 2009. Active Integrity Constraints for Database Consistency Maintenance. *IEEE Transactions on Knowledge and Data Engin.* 21, 7 (2009), 1042–1058.

[20] Luciano Caroprese, Irina Trubitsyna, Miroslaw Truszczynski, and Ester Zumpano. 2012. The View-Update Problem for Indefinite Databases. In *JELIA*. 134–146.

[21] Luciano Caroprese, Irina Trubitsyna, Miroslaw Truszczynski, and Ester Zumpano. 2014. A Measure of Arbitrariness in Abductive Explanations. *Theory and Practice of Logic Progr.* 14, 4-5 (2014), 665–679.

[22] Luciano Caroprese and Miroslaw Truszczynski. 2011. Active integrity constraints and revision programming. *Theory and Practice of Logic Progr.* 11, 6 (2011), 905–952.

[23] Stefano Ceri, Roberta Cochrane, and Jennifer Widom. 2000. Practical Applications of Triggers and Constraints: Success and Lingering Issues (10-Year Award). In *VLDB*. 254–262.

[24] Stefano Ceri, Piero Fraternali, Stefano Paraboschi, and Letizia Tanca. 1994. Automatic Generation of Production Rules for Integrity Maintenance. *ACM Trans. on Database Systems* 19, 3 (1994), 367–422.

[25] Stefano Ceri and Jennifer Widom. 1990. Deriving Production Rules for Constraint Maintainance. In *VLDB*. 566–577.

[26] Jan Chomicki. 2007. Consistent Query Answering: Five Easy Pieces. In *ICDT*. 1–17.

[27] Jan Chomicki, Jorge Lobo, and Shamim A. Naqvi. 2003. Conflict Resolution Using Logic Programming. *IEEE Transactions on Knowledge and Data Engineering* 15, 1 (2003), 244–249.

[28] Alvaro Cortés-Calabuig, Marc Denecker, Ofer Arieli, and Maurice Bruynooghe. 2006. Representation of Partial Knowledge and Query Answering in Locally Complete Databases. In *LPAR*. 407–421.

[29] Luís Cruz-Filipe. 2014. Optimizing Computation of Repairs from Active Integrity Constraints. In *FoIKS*, Christoph Beierle and Carlo Meghini (Eds.), Vol. 8367. Springer, 361–380.

[30] Luís Cruz-Filipe. 2016. Grounded Fixpoints and Active Integrity Constraints. In *ICLP*, Manuel Carro, Andy King, Neda Saeedloei, and Marina De Vos (Eds.), Vol. 52. 11:1–11:14.

[31] Luís Cruz-Filipe, Graça Gaspar, Patrícia Engrácia, and Isabel Nunes. 2013. Computing Repairs from Active Integrity Constraints. In *TASE*. IEEE Computer Society, 183–190.

[32] Bernardo Cuenca Grau, Ian Horrocks, Markus Krotzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang. 2013. Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. *J. Artif. Intell. Res.* 47 (2013), 741–808.

[33] James P. Delgrande, Torsten Schaub, and Hans Tompits. 2003. A Framework for Compiling Preferences in Logic Programs. *Theory and Practice of Logic Programming* 3, 2 (2003), 129–187.

[34] Alin Deutsch, Alan Nash, and Jeff B. Remmel. 2008. The Chase Revisisted. In *PODS*. 149–158.

[35] Guillaume Feuillade, Andreas Herzig, and Christos Rantsoudis. 2019. A Dynamic Logic Account of Active Integrity Constraints. *Fundam. Inform.* 169, 3 (2019), 179–210.

[36] Sergio Flesca, Filippo Furfaro, and Francesco Parisi. 2005. Consistent Query Answers on Numerical Databases Under Aggregate Constraints. In *DBPL*. 279–294.

[37] Sergio Flesca, Filippo Furfaro, and Francesco Parisi. 2007. Preferred Database Repairs Under Aggregate Constraints. In *SUM*. 215–229.

[38] Sergio Flesca, Filippo Furfaro, and Francesco Parisi. 2010. Range-Consistent Answers of Aggregate Queries under Aggregate Constraints. In *SUM*. 163–176.

[39] Sergio Flesca and Sergio Greco. 2001. Declarative semantics for active rules. *Theory and Practice of Logic Programming* 1, 1 (2001), 43–69.

[40] Filippo Furfaro, Sergio Greco, and Cristian Molinaro. 2007. A three-valued semantics for querying and repairing inconsistent databases. *Annals of Math. and Artificial Intelligence* 51, 2-4 (2007), 167–193.

[41] Ariel Fuxman and Renée J. Miller. 2005. First-Order Query Rewriting for Inconsistent Databases. In *ICDT*. 337–351.

[42] Ariel Fuxman and Renée J. Miller. 2007. First-order query rewriting for inconsistent databases. *J. Comput. System Sci.* 73, 4 (2007), 610–635.

[43] Gianluigi Greco, Sergio Greco, Irina Trubitsyna, and Ester Zumpano. 2005. Optimization of bound disjunctive queries with constraints. *Theory Pract. Log. Program.* 5, 6 (2005), 713–745.

[44] Gianluigi Greco, Sergio Greco, and Ester Zumpano. 2003. A Logical Framework for Querying and Repairing Inconsistent Databases. *IEEE Trans. on Knowledge and Data Engin.* 15, 6 (2003), 1389–1408.

[45] Sergio Greco and Cristian Molinaro. 2007. Querying and Repairing Inconsistent Databases Under Three-Valued Semantics. In *ICLP*. 149–164.

[46] Sergio Greco and Cristian Molinaro. 2008. Approximate Probabilistic Query Answering over Inconsistent Databases. In *ER*. 311–325.

[47] Sergio Greco and Cristian Molinaro. 2012. Probabilistic query answering over inconsistent databases. *Annals of Mathematics and Artificial Intelligence* 64, 2-3 (2012), 185–207.

[48] Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. 2013. Logic programming with function symbols: Checking termination of bottom-up evaluation through program adornments. *Theory Pract. Log. Program.* 13, 4-5 (2013), 737–752.

[49] Sergio Greco, Cristian Molinaro, and Irina Trubitsyna. 2018. Computing Approximate Query Answers over Inconsistent Knowledge Bases. In *IJCAI*, Jérôme Lang (Ed.). ijcai.org, 1838–1846.

[50] Sergio Greco, Cristian Molinaro, Irina Trubitsyna, and Ester Zumpano. 2010. NP Datalog: A logic language for expressing search and optimization problems. *Theory Pract. Log. Progr.* 10, 2 (2010), 125–166.

[51] Sergio Greco, Cristina Sirangelo, Irina Trubitsyna, and Ester Zumpano. 2003. Preferred Repairs for Inconsistent Databases. In *IDEAS*. 202–211.

[52] Sergio Greco, Cristina Sirangelo, Irina Trubitsyna, and Ester Zumpano. 2004. Feasibility Conditions and Preference Criteria in Querying and Repairing Inconsistent Databases. In *DEXA*, Vol. 3180. 44–55.

[53] Sergio Greco, Irina Trubitsyna, and Ester Zumpano. 2006. On the Semantics of Logic Programs with Preferences. In *JELIA (Lecture Notes in Computer Science, Vol. 4160)*, Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa (Eds.). Springer, 203–215.

[54] Sergio Greco, Irina Trubitsyna, and Ester Zumpano. 2007. On the Semantics of Logic Programs with Preferences. *Journal of Artificial Intelligence Research* 30 (2007), 501–523.

[55] Sergio Greco and Ester Zumpano. 2000. Querying Inconsistent Databases. In *LPAR*. 308–325.

[56] Bruno Marnette. 2009. Generalized schema-mappings: From termination to tractability. In *PODS*. 13–22.

[57] Wolfgang May and Bertram Ludascher. 2002. Understanding the global semantics of referential actions using logic rules. *ACM Transactions on Database Systems* 27, 4 (2002), 343–397.

[58] Claudia Bauzer Medeiros and Márcia Jacobina Andrade. 1994. Implementing integrity control in active data bases. *Journal of Systems and Software* 27, 3 (1994), 171–181.

[59] Luigi Palopoli and Riccardo Torlone. 1997. Generalized Production Rules as a Basis for Integrating Active and Deductive Databases. *IEEE Transactions on Knowledge and Data Engineering* 9, 6 (1997), 848–862.

[60] Norman W. Paton and Oscar Diaz. 1999. Active Database Systems. *Comput. Surveys* 31, 1 (1999), 63–103.

[61] Christos Rantsoudis, Guillaume Feuillade, and Andreas Herzig. 2017. Repairing ABoxes through Active Integrity Constraints. In *DL (CEUR Workshop Proceedings, Vol. 1879)*, Alessandro Artale, Birte Glimm, and Roman Kontchakov (Eds.). CEUR-WS.org.

[62] Chiaki Sakama and Katsumi Inoue. 2000. Prioritized logic programming and its application to commonsense reasoning. *Artificial Intelligence* 123, 1-2 (2000), 185–222.

[63] V. S. Subrahmanian. 1994. Amalgamating Knowledge Bases. *ACM Transactions Database Systems* 19, 2 (1994), 291–331.

[64] Jennifer Widom and Stefano Ceri (Eds.). 1996. *Active Database Systems: Triggers and Rules For Advanced Database Processing*. Morgan Kaufmann.

DC-SMIL: a Multiple Instance Learning Solution Via Spherical Separation for Automated Detection of Displastyc Nevi

Eugenio Vocaturo
DIMES, University of Calabria
CNR-NANOTEC, National Research Council
87036 Rende (CS), Italy
e.vocaturo@dimes.unical.it;eugenio.vocaturo@cnr.it

Ester Zumpano
DIMES, University of Calabria
87036 Rende (CS), Italy
e.zumpano@dimes.unical.it

Giovanni Giallombardo
DIMES, University of Calabria
87036 Rende (CS), Italy
giovanni.giallombardo@unical.it

Giovanna Miglionico
DIMES, University of Calabria
87036 Rende (CS), Italy
g.miglionico@dimes.unical.it

ABSTRACT

Among skin cancers, melanoma is the most aggressive and most lethal form. Despite these terrible premises, an excision treatment carried out thanks to an early diagnosis is almost always decisive, guaranteeing the patient's survival. The early detection of melanoma is hampered by the extreme similarity of melanoma with other skin lesions such as dysplastic nevi. The current research is aimed at defining software solutions that support the computerized diagnosis of lesions for the detection of melanoma. To date, the proposals, both in terms of algorithms and frameworks, have focused on the dichotomous distinction of melanoma from benign lesions. However, the current debate on *Dysplastic Nevi Syndrome (DNS)*, makes issues relating to the nature of the lesions, central to subjects who present a large number of moles throughout the body. In fact, individuals with DNS have a greater chance of being attacked by melanoma. The classification task relating to the distinction of dysplastic nevi from common ones is totally unexplored. In this document, we consider the difficult task of applying multiple-instance learning (MIL) approaches to discriminate melanoma from dysplastic nevi and outline an even more complex challenge related to the classification of dysplastic nevi from common ones. In particular, we introduce the application of a MIL approach that uses spherical separation surfaces. Since the results seem promising, we conclude that a MIL technique could be the basis of more sophisticated tools useful for detecting skin lesions.

CCS CONCEPTS

• **Information systems** → *Expert systems*; *Data management systems*; • **Applied computing** → **Health informatics**;

KEYWORDS

Dermoscopy imaging Classification, Melanoma, Dysplastic Moles, Spherical Multiple Instance Learning

ACM Reference Format:
Eugenio Vocaturo, Ester Zumpano, Giovanni Giallombardo, and Giovanna Miglionico. 2020. DC-SMIL: a Multiple Instance Learning Solution Via Spherical Separation for Automated Detection of Displastyc Nevi. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3410566.3410592

1 INTRODUCTION

In [1] the World Health Organization reports that in 2018 more than 60.000 persons died due to melanoma and the new cases are over 280.000. Melanoma is currently the fifth type of cancer for deaths in the world, and is the most deadly type of skin cancer (Fig. 1 and Fig. 2).

Anyhow, melanoma is curable if it is recognized with and early diagnosis. In order to identify the initial stadium of the lesions, different clinical protocol exist.

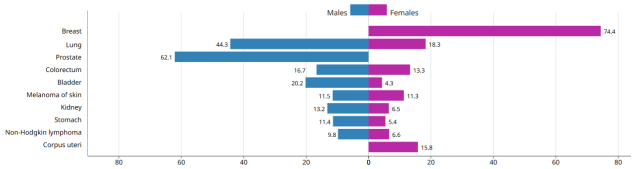


Figure 1: Incidence rates per sex, top 10 cancers [1]

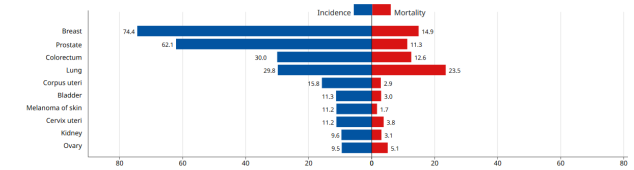


Figure 2: Incidence and mortality rates, top 10 cancers [1]

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410592

The basis idea common to all them is supporting the specialists in melanoma identification by examining and keep trace of each evolution over time of some specific features such as irregular edges, diameters greater than 6 mm, asymmetry and color. In order to support the early detection of melanoma, Computer Aided Diagnosis (CAD) systems have been developed by the scientific community. These tools guarantee an automatic analysis of the lesion, and give evidence to some specific features suggested by the medical team, so that ensuring an effective contribution to face melanoma. CAD steps usually include image acquisition, pre-processing, segmentation, features extraction and selection and finally classification of lesions.

Each step is challenging and has to be correctly performed for the entire process to be effective. Regarding the image acquisition, it is increasingly adopted the use of advanced techniques such as imaging with dermatoscopy, also known as *epiluminescence microscopy* (ELM), which allows much more detailed images. This contribution relates to the classification tasks of discriminating melanoma from dysplastic nevi and dysplastic nevi from common ones.

Previous results evidence that specific ethnic groups present on their bodies surface a great number of common and dysplastic nevi (e.g. [2] reports that 8% of the Caucasian population have dysplastic nevi or unusual lesions that may resemble melanoma). Even if, just a few number of dysplastic nevi degenerate into a melanoma [3] it is evident that greater risk of developing melanoma pertains to individuals with *dysplastic nevi syndrome* or with family history of melanoma.

Bottom line, we argue and this motivates this research, that automatic diagnosis besides being relevant in the distinction between melanoma and common nevi. This work is a contribution in the challenging field of melanoma detection. More specifically our research focuses on dermoscopic images and proposes the application of DC-SMIL [6], a multiple instance learning algorithm, on the challenging tasks of classifying:

- melanoma vs dysplastic nevi
- dysplastic nevi vs common ones.

The first task is difficult for the great similarity of the two types of lesions [4]. Even more complex is the classification of dysplastic nevi from common ones: this issue is completely new and has not been addressed in the literature. These claims are true both in the case of a traditional diagnosis made by physicians both in the case of classifier algorithms. Our goal is to verify how the MIL approaches are interesting when applied on binary classification tasks in which the images are very similar to each other. How possible development we will consider the two tasks of binary classification addressed in the present paper as a multi-classification problem. In this sense, it is possible to combine several binary classifiers into one single multi-class classifier using either the One-versus-All (OvA) or the All-versus-All (AvA) approach [5].

The paper is organized as follows. In the next section we focus on the role that the presence of dysplastic nevi and common nevi may imply in terms of risk of melanoma onset. In Section 3 we introduce the Multiple Instance Learning approach, and in section 4 we focus on DC-SMIL e new MIL algoritmh that adopt spherical separation surfaces [6]. In section 5 we have recalled the classification indices we have referred to, and in Section 6 we describe the datasets used to test DC-SMIL. Finally, in section 7, we report some numerical results on literature datasets and on dermoscopic dataset, and in section 8 some conclusions are given.

2 DYSPLASTIC NEVI

The term “dysplastic nevus” (DN) derives from the Greek “dis-” (bad or malfunction) and “-plasia” (development of growth or change) [7], and it indicates a potentially dangerous lesion for the individual denoting a nevus presenting different histological and genetic characteristics compared to common nevus. Many different works have studied the correlation between dysplasia of melanocytes with the risk of melanoma [8–10], but at the present no unique proved vision has been given by the scientific community. A more general debate exists in the scientific community, to well define the relationships related to the presence of dysplastic and common nevi and the possibility of melanoma occurrence [2].

The *Syndrome of Dysplastic Nevus (DNS)* refers to individuals that present a high number of both benign moles and dysplastic nevi. Fortunately, just a small percentage of them will develop a melanoma formation [11]. Individuals with dysplastic nevi are more likely to develop a malignant transformation if melanoma familiarly conditions exists.

In [12], a cumulative lifetime risk of almost 100% is reported for individuals who have dysplastic nevi and are related to melanoma; about 30% of melanomas occur within atypical moles. A genetic predisposition for the formation of melanoma is present in 40-50% of cases. The correlation between the presence of dysplastic nevi and the melanoma has been also investigated in [13]. The diagnosis of a severe DNS cannot be overlooked, as it could state for a miss-diagnosed in situ melanoma [14], it may reflect the dermatopathological uncertainty related to a wrong diagnosis. Figure 3 reports a dermoscopic image of common nevi, dysplastic nevi and melanoma.

Figure 3: Dermoscopic image of common nevus (a), dysplastic nevus (b) and melanoma (c)

Basically, the risk of melanoma is related to two different objective criteria:

- An increased risk of melanoma is related to a high number of nevi [15]. In [16] reports that individuals with a number of nevi greater than 100 have a risk of melanoma 7 times greater than those with a count of less than 15.
- An increased risk of melanoma is related to the presence of large nevi. A histological study of nevi has shown that higher is the extension of the mole, greater is the risk of turning into melanoma: the relative risk of 1 for nevi with a diameter less than 2.4 mm, while the relative risk progressively increases up to 5 if the lesion has a diameter greater than 4.4 mm [17].

Various studies on the exact cause-effect correlations have been provided over time, as well as solutions for the automatic identification of skin lesions.

In [18], the authors analyze the performances of various proposals, taking into account the particular value that sensitivity, specificity and dataset size may have in the medical context.

In general, measures such as accuracy or F-score are commonly used to determine the quality of the classification. In the same work methods are categorized based on their classification scope: melanoma from benign (M *vs* B), melanoma from benign and dysplastic (M *vs* $(B + D)$) and melanoma versus only dysplastic nevi (M *vs* D).

The comparison among different approaches is far from easy. The motivation is that each proposal has been applied on different datasets and moreover adopts different features sets. For this last an additional difference arises between global and local features.

Global features are extracted taking the lesion as a whole, whereas local features are extracted from portions of the image. As for a comparison between global and local features we point out that the local approach allows to increase the size of the features vector, but also the complexity of the features space. Different approaches have been proposed in the literature to manage these drawbacks: the bag of features (BoF) approach in [40], [19], together with the use of MIL approaches that simplify the training set annotation phase and allow to make a less expensive image segmentation step [20–23]. Reasoning on a qualitative level the most effective methods appear to be: AdaBoost (AdB), artificial neural network (ANN), Support Vector Machines (SVM).

Anyhow, fewer attentions have been given to the discrimination of melanoma from dysplastic nevi. The topic investigated in this paper is the classification task of dysplastic nevi against common nevi, which to the best of our knowledge has never been taken into consideration.

3 MULTIPLE INSTANCE LEARNING

Machine Learning has become very important in medical image analysis. In fact, machine learning methods are currently used in the segmentation steps, in which each pixel of an image belongs to a particular tissue and in CAD systems to assign a category label to a whole image. Even, the availability of partially labeled data can be appropriately exploited using machine learning approaches [24], [25], [26]. Manual labeling of images is a time-consuming activity, and may not be strictly necessary in clinical practice. On the other hand, even when labeled data is collected, they are often not available for all researchers. To overcome these problems, approaches such as semi-supervised learning, multi-instance learning and transfer learning have become popular.

Multiple Instance Learning scenario is particularly useful when disposing of local annotated labels is expensive, while global labels for whole images, such as the outcome of a diagnosis, are more readily available. MIL is an extension of supervised learning that can train classifiers using weakly labeled data. The goal is therefore to exploit the labels of the weaker bags for training. A MIL problem consists in the classification task of a set of items called *bags* and of the objects inside them called *instances*. The substantial difference compared to supervised classification consists in the fact that, in the learning phase, only the labels of the bags are known, and not those of the instances.

In section 7, to better discuss the results provided by the DC-SMIL algorithm, we will make a comparison with the results obtained with two other recent MIL algorithms, namely MIL-RL [41] and DC-MIL [42]. In [41] the authors start from a mixed integer nonlinear optimization model drawn from the literature proving that a Lagrangian relaxation approach, equipped with a dual ascent scheme, allows to obtain an optimal solution of the original problem. In [42] adopting a support vector machine setting at the training level, the problem of minimizing the classification- error function has been formulated as a nonconvex nonsmooth unconstrained program. The authors here proposed a difference-of-convex (DC) decomposition of the nonconvex function using an appropriate nonsmooth DC algorithm.

The MIL paradigm is particularly well suited to image classification, given that to classify an image containing an object of interest, it is necessary to examine only some sub-regions of the same image. With a MIL approach it is therefore possible to obtain global information from local one. For further details and general considerations on the MIL paradigm, we refer the reader to surveys [27, 28]. In [20], a detailed review is given concerning Multiple Instance Learning applied for medical images and video analysis. The MIL approach, as far as we know, is still very rarely used for melanoma detection, and has never been used for the detection of dysplastic nevi.

4 MULTIPLE INSTANCE CLASSIFICATION VIA SPHERICAL SEPARATION

In this section we describe the heuristics optimization MIL algorithm, named DC-SMIL, proposed in [6], which is suitable for image classification.

DC-SMIL adopt spherical separation as a classification tool and come out with an optimization model which is of DC (Difference of Convex) type. The model was addressed using a specialized nonsmooth optimization algorithm, recently proposed in the literature which is based on objective function linearization and bundling. In particular the adopted classification error function depend on center and radius of the sphere and we come out with an optimization model to minimize a combination of the volume of the sphere and of the classification error. Through a DC (Difference of Convex) decomposition of the objective function it is possible to resort to a number of effective algorithms available in the literature, see [29–33]. In subsection 4.1, to facilitate understanding of the approach, we introduce the spherical separation model.

4.1 Problem statement

We assume that a set of instances $X = \{x_1, \ldots, x_N\}$ is given in the sample space $\mathbb{R}^n$, which is partitioned into $m + k$ subsets, named *bags*, $X_1^+, \ldots, X_m^+, X_1^-, \ldots, X_k^- \subset X$. Hence, each bag is constituted by a set of instances (i.e., points in the sample space $\mathbb{R}^n$), and each instance belongs to exactly one bag.

The subsets $X_1^+, \ldots, X_m^+$ are referred to as the *positive bags*, while $X_1^-, \ldots, X_k^-$ are referred to as the *negative bags*. We denote by $J_1^+, \ldots, J_m^+$ the instance index-sets of the positive bags $X_1^+, \ldots, Y_m^+$, by $J_1^-, \ldots, J_k^-$ the instance index-sets of the negative bags $X_1^-, \ldots, X_k^-$,

and we let

$$J^+ \triangleq \{J_1^+, \ldots, J_m^+\} \qquad \text{and} \qquad J^- \triangleq \{J_1^-, \ldots, J_k^-\}.$$

As stated earlier, our aim is to find a sphere $S(w,r) \subset \mathbb{R}^n$, of center $w \in \mathbb{R}^n$ and radius $r \in \mathbb{R}$, separating the two classes of bags. In the following definition we state that in order to separate the positive bags $X_1^+, \ldots, X_m^+$ from the negative ones $X_1^-, \ldots, X_k^-$, a sphere must have a nonempty intersection with each positive bag, while leaving outside all the instances belonging to negative bags.

Definition 4.1. Let a sphere $S(w,r)$ of center $w \in \mathbb{R}^n$ and radius $r \in \mathbb{R}$ be given. $S(w,r)$ is called a *separating* sphere if for every $i \in \{1, \ldots, m\}$ it holds:

$$\|x_j - w\|^2 - r^2 \leq 0 \qquad \text{for some } j \in J_i^+ \tag{1}$$

and for every $i \in \{1, \ldots, k\}$ it holds:

$$\|x_j - w\|^2 - r^2 \geq 0 \qquad \text{for every } j \in J_i^-. \tag{2}$$

A pictorial example of spherical separation is presented in Figure 4, where the sphere $S(w,r)$ separates the negative bags X_1^-, X_2^-, and X_3^- from the positive bags X_1^+ and X_2^+. In particular, we remark that while the bags depicted in Figure 4 are spherically separable, they are not separable by any hyper-plane. According to Definition 4.1

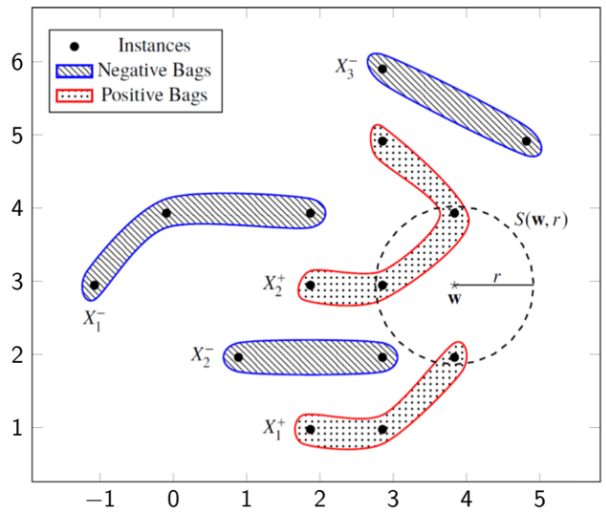


Figure 4: Spherical separation with three negative bags and two positive bags [6]

any negative bag X_i^-, with $i \in \{1, \ldots, k\}$, is said *misclassified* with respect to a given sphere $S(w,r)$ if there exists $j \in J_i^-$ such that $r^2 - \|x_j - w\|^2 > 0$. Likewise, any positive bag X_i^+, with $i \in \{1, \ldots, m\}$ is said *misclassified* with respect to $S(w,r)$ if $\|x_j - w\|^2 - r^2 > 0$ for every $j \in J_i^+$.

Based on the latter remark an optimization model was obtained whit the aim to look for a separating sphere, if any, by minimizing a measure of all the classification errors of both the negative and the positive bags.

The resulting problem P and the loss function f can be written as:

$$\min_{(w,r)\in\mathbb{R}^{n+1}} f(w,r) \tag{3}$$

where

$$f(w,r) \triangleq r^2 + C \sum_{i=1}^{k} \max\left\{0, \max_{j\in J_i^-}\left\{r^2 - \|x_j - w\|^2\right\}\right\} + C \sum_{i=1}^{m} \max\left\{0, \min_{j\in J_i^+}\left\{\|x_j - w\|^2 - r^2\right\}\right\} \tag{4}$$

Then our loss function f is given by the sum of three terms:

- by the first term we would like to minimize the volume of the sphere;
- by the second one we would like to minimize the missclassification error of the negative bags;
- while by the last term we would like to minimize the missclassification error of the positive bags.

With respect to the decision-variable vector $(w,r) \in \mathbb{R}^{n+1}$ and the related sphere $S(w,r)$, we define for every negative bag X_i^-, with $i \in \{1, \ldots, k\}$, the classification error $\mathcal{E}_i^-(w,r)$ as:

$$\mathcal{E}_i^-(w,r) \triangleq \max\left\{0, \max_{j\in J_i^-}\left\{r^2 - \|x_j - w\|^2\right\}\right\},$$

and for every positive bag X_i^+, with $i \in \{1, \ldots, m\}$, the classification error $\mathcal{E}_i^+(w,r)$ as:

$$\mathcal{E}_i^+(w,r) \triangleq \max\left\{0, \min_{j\in J_i^+}\left\{\|x_j - w\|^2 - r^2\right\}\right\}.$$

Putting together the classification errors of all the positive and negative bags, we obtain the following spherical MIL error function $\mathcal{E}(w,r)$

$$\mathcal{E}(w,r) = \sum_{i=1}^{k} \mathcal{E}_i^-(w,r) + \sum_{i=1}^{m} \mathcal{E}_i^+(w,r), \tag{5}$$

and we note that $\mathcal{E}(w,r) \geq 0$, where $\mathcal{E}(w,r) = 0$ if and only if $S(w,r)$ is a separating sphere according to Definition 4.1.

So the Spherical MIL problem (SMIL) descends as the following unconstrained optimization problem

$$\min_{(w,r)\in\mathbb{R}^{n+1}} f(w,r) \triangleq r^2 + C\mathcal{E}(w,r), \tag{6}$$

which combines, by introducing a trade-off parameter $C > 0$, the two objectives of minimizing the radius of the sphere and the classification errors of all the negative and positive bags. Here the radius minimization is aimed at reducing the false positive phenomenon when the calculated sphere is used as a classification tool.

We remark that the error function $\mathcal{E}(w,r)$ is difficult to be minimized because it is inherently nonconvex and nonsmooth. Also the function $f(w,r)$ is difficult to be minimized but has a good propriety: it is possible to introduce a decomposition of $f(w,r)$ as the difference of two convex nonsmooth functions that will allow us to tackle the problem by adopting nonsmooth DC algorithms of the type described in [31], where the authors introduced a proximal bundle method for the numerical minimization of a nonsmooth difference-of-convex (DC) function.

The proposed method iteratively build two separate piecewise-affine approximations of the component functions, grouping the corresponding information in two dedicated bundles. In the bundle of the first component, only information related to points close to the current iterate are maintained, while the second bundle only refers to a global model of the corresponding component function. By combining the two convex piecewise-affine approximations, it is possible to generate a DC piecewise-affine model, which can also be seen as the pointwise maximum of several concave piecewise-affine functions. Such a nonconvex model is locally approximated by means of an auxiliary quadratic program, whose solution is used to certify approximate criticality or to generate a descent search-direction, along with a predicted reduction, that is next explored in a line-search setting.

To validate the DC-SMIL method we have adopted the DC Piecewise - Concave Algorithm (DCPCA) for minimizing a nonsmooth DC function as introducted in [31]. For further details please see [6].

5 CLASSIFICATION INDEX

The formal definition of accuracy, sensitivity and specificity derives from the quantification of the number of true positives (TP), true negatives (TN), false negatives (FN) and false positives (FP).

If a disease is actually present in a patient and the diagnostic test detects the disease, the test result is considered true positive (TP). Similarly, if a patient is healthy and the diagnostic test does not detect the disease, the test result is true negative (TN). However, if the diagnostic test indicates the presence of a disease in a healthy patient, the test result is false positive (FP). Similarly, if the diagnostic test result does not detect the disease in a sick patient, the test result is false negative (FN).

The measures we used to evaluate the methods considered are:

$$Correctness = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{7}$$

$$Sensitivity = \frac{TP}{TP + FN} \times 100\% \tag{8}$$

$$Specificity = \frac{TN}{TN + FP} \times 100\% \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{10}$$

$$Recall \equiv Specificity \tag{11}$$

$$F_{Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

$$CPU_{time} = \frac{CPU\ \ clock\ \ cycles}{clock\ \ frequency} \tag{13}$$

Correctness (7), also referred as "accuracy", specifies if a model is properly trained and how it can work in general. The problem with using correctness as the main performance metric is when the classes are significantly not balanced.

Sensitivity and specificity are the most commonly used performance evaluation parameters in the literature, at least in medical field. *Sensitivity* (8), is a measure of correctly identifying non-healthy patients. *Specificity* (9) or "recall", is a measure of the capability to identify healthy patients (i.e., to avoid false positive patients). *Precision* (10) indicates how much were correctly classified as positive out of all positives.

To optimize the evaluation metric, *F-score* (12) is frequently used, which is a combined measure of recall (specificity) and precision. A good F-score means we have low false positives and low false negatives, so you are carefully identifying real threats and not being bothered by false alarms. F-score is considered perfect when it is 1, while the model fails when it is 0.

CPU time (13), or "process time", is the amount of time for which a central processing unit (CPU) is used for processing instructions of a computer program. The CPU time is measured in clock ticks or seconds. CPU time may decrease both by increasing the frequency of the clock or by decreasing the clock cycles needed to run the program.

Regarding the classification performance of skin lesions, 5 and 10 fold cross validation are almost always adopted. The *k-fold-cross validation* consists in dividing, for k times, all the data into k pieces: $k-1$ pieces are used for training and the last piece as a test set. At the end of this process the average test correctness is taken. If the original data set is too small, the recommended strategy is to use *leave one out cross-validation (LOO CV)* that by taking the k-fold-cross validation approach to extremes, allow to compare each single image with all the others [34]. Practically, when k coincides with the cardinality of the entire dataset, the *k-fold cross validation* is called *leave-one-out cross validation*: in such case, each time the testing set is constituted just by one element.

6 DATA SETS

We have performed experiments applying DC-SMIL on various data sets to evaluate the goodness of the proposed technique and to compare the obtained results with those of other methods for MIL. In particular, we performed some numerical experiments by testing it on:

- five medium-size benchmark problems,
- and on seven large size benchmark problems,
- a dermatoscopic dataset (PH_2).

A brief description of the datasets is reported in the following paragraph. The first two data sets are used to evaluate the performance of the proposed MIL algorithm; in particular Elephant, Fox and Tiger are the reference benchmark datasets used for MIL image recognition problems.

In light of the numerical results on the reference datasets, we applied DC-SMIL on a real dermatoscopic dataset (PH_2), with the aim of verifying that the MIL spherical separation approach may be of interest in classification tasks in which the data to be classified have extreme similarity. PH_2 is a medical dataset used to test the classification performances of machine learning approaches for classification task on melanoma detection [35].

6.1 Musk Data Sets

The MUSK data sets are the benchmark data sets used for testing in virtually all previous MIL approaches and have been described in detail in [36]. Both data sets, *Musk1* and *Musk2*, consist of descriptions of molecules that use multiple low-energy conformations. In these data sets each conformation deriving from surface properties is represented by a vector of 166 dimensional features. More precisely, *Musk1* contains on average about 6 conformations for each molecule, while *Musk2* has on average more than 60 conformations inside each bag.

Function vectors describing each conformation have been extracted for both Musk 1 and Musk 2. During the training of a classifier on these data sets, the classifier will indicate the *musk* class for a generic molecule if any of its conformations is classified as musk, viceversa a molecule will be classified as *non musk*.

6.2 Elephant, Fox and Tiger

One of the most important problems in computer vision is retrieving images from large data set using the image content as a search criterion. In [37] a new MIL data set was generated for an image annotation task. This data set was derived from original data consisting of color images taken from Corel data set.

In the seminal work [38] the Blobworld system was presented which represents images by applying a transformation from raw pixel data to a subset of image regions characterized by similar color and texture. Using the pre-processing and segmentation techniques described in [38], in [37] each image was decomposed into a set of segments characterized by color, texture and shape descriptors.

The three categories *Elephant, Fox, Tiger* were randomly extracted from a pool of photos of other animals, each of which characterized by 100 positive and 100 negative example images.

The reduced accuracy of image segmentation, the small number of region descriptors and the small size of the training set, make these data sets very difficult due to a classification problem. This justifies why these categories are generally used to test MIL classification algorithms.

6.3 PH_2 Data set

This dataset was set up by the Universidad do Porto and Tecnico Lisboa, in collaboration with the Dermatology Service of the Hospital Pedro Hispano (Portugal) [35]. These images have been classiiňĄed by expert dermatologists considering the manual segmentation of the skin lesion, the clinical and histological diagnosis and dermoscopic criteria like ABCDE protocol [39] looking at (asymmetry, colors, pigment network, particular structures). In our experiments, none of the features resulting from the manual analysis was used in the automatic classiiňĄcation process.

The entire PH_2 database contains 200 images of melanocytic lesions: 80 common nevi, 80 atypical nevi and 40 melanomas. All images were obtained using 8-bit RGB colors with a resolution of 768×560 pixels. The patients from whom the photos were taken correspond to the skin type II or III, according to Fitzpatrick's skin type classification scale: for this reason, the background color varies from white to cream white. These images have been classified by expert dermatologists based on the following parameters:

- manual segmentation of the skin lesion;
- clinical and histological diagnosis;
- dermoscopic criteria (asymmetry, colors, pigment network, particular structures).

For the classification experiments we considered the 40 images of melanomas (Fig. 5a), the 80 of dysplastic nevi (Fig. 5b) and the 80 of common nevi (Fig. 5c), without taking into account the indications resulting from the manual analysis carried out by the specialists.

For the experiments reported in the following section, we have considered respectively:

- positive the images related to melanomas and negative the ones related to dysplastic nevi;
- positive the images related to dysplastic nevi and negative the ones related to common nevi.

In [40] the authors demonstrated how by adopting only color features, satisfactory classification performances can be obtained using dermatoscopic images,. In our experimental section, we used an n-dimensional representation of the subregions of each image. In [23], we have divided each image (bag) into square subregions (instances). For each central sub-regions we have computed the following quantities:

- The average of the RGB channels intensities of the subregion (3 features);
- The variance of the RGB channels intensities of the subregion (3 features);
- The differences between the average of the RGB channels intensities of the subregion and that ones of the upper, lower, left, right adjacent subregions (12 features);
- The differences between the variance of the RGB channels intensities of the subregion and that ones of the upper, lower, left, right adjacent subregions (12 features).

Then we come out with 30 color features for each instance. For further details please see [23]. In the following experimental section, we used a 30-dimensional representation for each sub-regions of each image, considering this same representation for all binary classifiers.

7 NUMERICAL RESULTS AND FINAL REMARKS

We have assessed the performance of DC-SMIL, by testing it on a set of five medium-size benchmark problems extracted from [36, 37], and on a dermatoscopic dataset [35].

The relevant characteristics of each problem are reported in Table 1. In this table we list the problem size n (i.e., the number of features), the number of instances N, the number of positive bags m, and the number of negative bags k.

In Table 2 we report the results provided by DC-SMIL in terms of average test-correctness compared with those of MIL-RL[41], DC-MIL [42] and mi-SVM* as presented in [41]. In [41], in particular, the authors applied to image classiiňĄcation an instance-level technique based on the Lagrangian relaxation of a Support Vector Machine (SVM) type model obtaining interesting numerical results. These algorithms were implemented by our research group, using exactly the same folds in the 10-cross fold validation.

We can see that DC-SMIL is the best performance method on 4 out of 5 test problems.

Data sets	n	N	m	k
Elephant	230	1391	100	100
Fox	230	1320	100	100
Tiger	230	1220	100	100
Musk 1	166	476	47	45
Musk 2	166	6598	39	63
PH_2	30	640	80	80

Table 1: Characteristics of Size Data sets

	10-CV			
	DC-SMIL	DC-MIL	MIL-RL	$mi-SVM^*$
Elephant	84.0	84.0	83.00	82.50
FoX	59.0	57.0	54.5	56.5
Tiger	77.5	84.5	75.0	77.5
Musk 1	80.0	74.5	80.0	76.7
Musk 2	80.0	74.0	73.0	77.0

Table 2: Average test-correctness on Literature Datasets

To avoid the problems related to the use of datasets with unbalanced classes, we have duplicated all the images of melanomas, adding to the repeated ones a Gaussian noise with zero mean with variance equal to 0.0001, as in the [40]. In this way we obtained a balanced dataset containing three classes of data, Melanomas (M), Dysplastic Nevi (DN) and Common Nevi (N) each with 80 images.

For our experiments we have considered the following two data set configurations:

- 160 images: 80 Melanomas vs 80 Dysplastic Nevi;
- 160 images: 80 Dysplastic Nevi vs 80 Common Nevi.

For each data set configuration, we performed a ten fold cross-validation. The respective results are listed in Tables 3 and 4, where we report the average of correctness, sensitivity, specificity, F score and CPU time.

In order to appreciate the MIL classification paradigm, we report in the columns MIL-RL, SVM and SVM-RBF the results obtained using MIL-RL algorithm and standard SVM approach [43] with linear and RBF kernels, respectively.

For each data set configuration the best results in Tables 3 and 4 have been underlined.

7.1 Melanomas vs Dysplastic Nevi

From numerical experiments it emerges that, in general, MIL-RL overcomes DC-SMIL and SVM technique (with both linear and RBF kernels) in terms of accuracy and sensitivity. Whenever accuracy is not 100%, low specificity values are a consequence of high sensitivity values.

In medical fields, sensitivity plays a more important role than specificity since it is a measure of the ability to identify un-healthy patients. The F-score values show the good performance of the MIL approach in classifying melanoma from dysplastic nevi against the classic SVM technique.

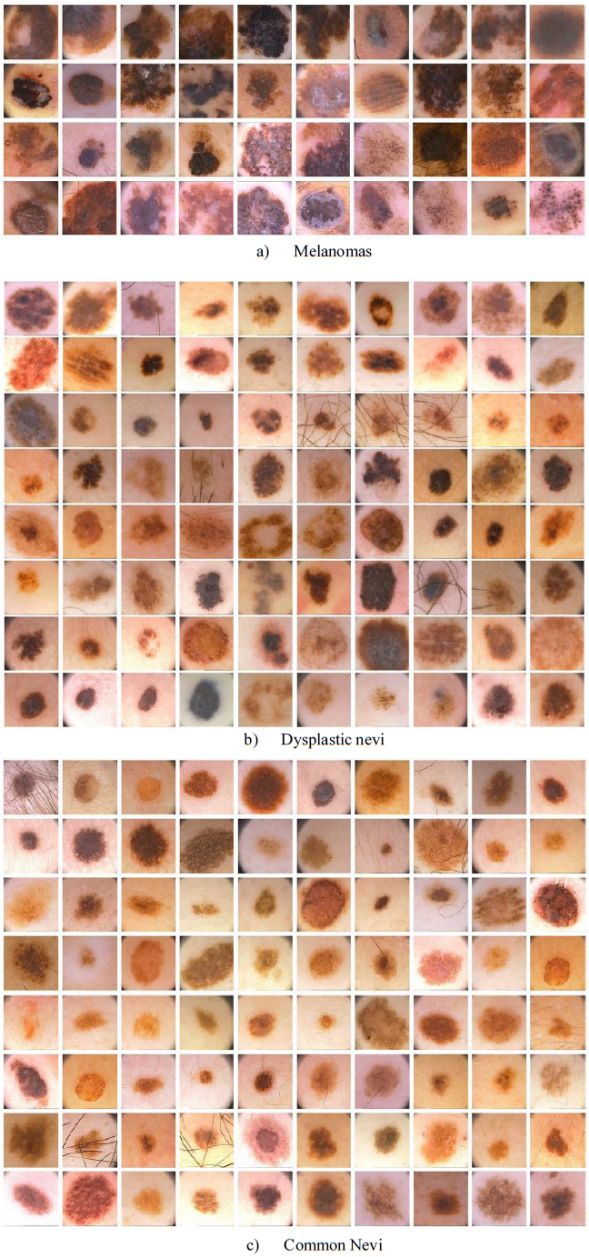


Figure 5: Dermoscopic image of Melanomas (a), dysplastic nevus (b) and common nevus (c)

7.2 Dysplastic Nevi vs Common Nevi

Several studies show that people who have a lot of dysplastic nevi have a greater chance of developing melanoma. Currently there is a debate in the scientific community, to well define the correlation related to the presence of dysplastic and common nevi and the possibility of melanoma occurrence [2]. Distinguish displastic nevi from common ones becomes an important task to diagnosis DNS (Displastic Nevi Syndrome).

	10-CV			
	DC-SMIL	MIL-RL	SVM	SVM-RBF
Correctness (%)	70.00	86.25	69.38	86.25
Sensitivity (%)	69.30	91,08	69.65	87.88
Specificity (%)	71.81	82.12	69.87	85.95
F-score (%)	69.09	87.01	68.68	87.52
CPU time (secs)	0.66	1.20	2.05	0.03

Table 3: 80 melanomas and 80 dysplastic nevi

	10-CV			
	DC-SMIL	MIL-RL	SVM	SVM-RBF
Correctness (%)	59.38	59.38	58.13	51.88
Sensitivity (%)	59.73	31.77	43.67	58.92
Specificity (%)	59.88	87.06	73.48	46.47
F-score (%)	57.61	42.77	48.57	53.74
CPU time (secs)	0.58	1.71	2.13	0.03

Table 4: 80 dysplatic nevi and 80 common nevi

With regard to the experimental section on the classification of dysplastic nevi against common nevi, the performances of MIL-RL and of SVM tecniques appear totally unsatisfactory. This is obvious because the images that were separated are very similar. MIL-RL registers the worst value of F-score and sensitivity, and overall it is not effective to solve the proposed task.

The use of spherical separating surfaces, provided by DC-SMIL algorithm, allows significant improvements in the extremely difficult task of classify dysplastic nevi from common ones. We conclude this section with some considerations about images characterized by the presence of a lot of hairs.

As shown in [44, 45] better results could be obtained in case of images pre-processing aimed at eliminating the presence of possible noises, such as possible hair or halo left by the dermoscopic gel used to allow better illumination of the lesions. Even the adoption of further useful features extracted from blob is a possibility that would allow to improve the classification performances [46–48]. Pre-processing steps and the adoption of a more numerous set of features appear to be an obligatory step when considering non-dermatoscopic images [49, 50].

8 CONCLUSION

In this paper we have presented an application of a multiple instance learning approach referring to the detection of melanoma by dysplastic nevi and of dysplastic nevi by common ones. These two processes are not widespread in the literature. Anyhow, pathologies such as the *Dysplastic Nevi Syndrome (DNS)* requires to implement tools that support phisicians in diagnostic process and mobile applications useful for promoting self diagnosis. To this end, we point out that it is under implementation a module of the software Simpatico 3d that is in charge of allowing self diagnosis [51, 52]. The obtained results show that in the first case MIL-RL is very promising, even in the conditions in which we performed the experiments, i.e. with only color features and without using pre-processing steps.

In the second case, MIL-RL algorithm as well as the SVM in the linear and Kernel RBF version, do not give satisfactory results. The excessive similarity of the lesions is not properly discriminated with approaches aimed at identifying linear separation surfaces. On the other hand DC-SMIL, thanks to the use of spherical separation surfaces, seems to be an interesting proposal for the development of applications in contexts in which positive and negative elements have similar characteristics. In these case the MIL paradigms adapt very well to the classifications of images because they are able to detect global information (bags) working locally (istance level).

Future research could include the design of more sophisticated segmentation techniques in order to further improve classification results, as well as the application of the proposed method in other medical fields [53, 54] to identify other types of injuries.

REFERENCES

[1] http://gco.iarc.fr/today/explore.
[2] Silva JH, de Sa B, Avila A, Landman G, Duprat Neto JP., "Atypical mole syndrome and dysplastic nevi: identification of populations at risk for developing melanoma," Clinics (Sao Paulo), v.66(3):493âĂŞ9, 2011.
[3] A.C. Society, Melanoma skin cancer. http://www.cancer.org/acs/groups/cid/documents/webcontent/003120-pdf.pdf
[4] Burroni M, Sbano P, Cevenini G, Risulo M, DellâĂŹEva G, Barbini P, et al. "Dysplastic naevus vs. in situ melanoma: digital dermoscopy analysis". Br J Dermatol, v.152(4), pp:679âĂŞ84, 2005.
[5] , Aly M.,"Survey on multiclass classification methods", Neural Netw, v. 19, pp. 1-9, 2005.
[6] Gaudioso, M., Giallombardo, G., Miglionico, G., and Vocaturo, E., "Classification in the multiple instance learning framework via spherical separation". Soft Computing, pp:1-7, 2019.
[7] Duffy K, Grossman D., "The dysplastic nevus: from historical perspective to management in the modern era: part I. Historical, histologic, and clinical aspects". J Am Acad Dermatol., 67(1):1.e1-1.e16. quiz 17-8, 2012.
[8] Greene, M. H. and Clark Jr, Wallace H. and Tucker, M. A. and Kraemer, K. H. and Elder, D. E. and Fraser, M. C., "High risk of malignant melanoma in melanoma-prone families with dysplastic nevi". Annals of internal medicine, 102(4), pp: 458-465, 1985.
[9] Jeong, D. K., Bae, Y. C., Lee, S. J., Kim, H. S., and Choi, Y. J., "A case of malignant melanoma after repeated recurrent dysplastic nevi". Archives of craniofacial surgery, 20(4), 260, 2019.
[10] Save, S., "Dysplastic Nevi", Dermoscopy: Text and Atlas, 447, 2019.
[11] Marghoob, A., and Braun, R.,"An atlas of dermoscopy". CRC Press, 2012.
[12] R. Pampena, A. Kyrgidis, A. Lallas, E. Moscarella, G. Argenziano, C. Longo,"A meta-analysis of nevus-associated melanoma: Prevalence and practical implications". In: Journal of the American Academy of Dermatology. Band 77, Nummer 5, pp. 938âĂŞ945, 2017.
[13] Arumi-Uria M, McNutt NS, Finnerty B., "Grading of atypia in nevi: correlation with melanoma risk". Mod Pathol., 16(8):764-771, 2003.
[14] Reddy KK, Farber MJ, Bhawan J, Geronemus RG, Rogers GS., "Atypical (dysplastic) nevi: outcomes of surgical excision and association with melanoma". JAMA Dermatol.,149(8): 928-934, 2013.
[15] Rieger E, Soyer HP, Garbe C, et al., "Overall and site-specific risk of malignant melanoma associated with nevus counts at different body sites: a multicenter case-control study of the German Central Malignant-Melanoma Registry". Int J Cancer. 62(4):393-397, 1995.
[16] Gandini S, Sera F, Cattaruzza MS, Pasquini P, Abeni D, Boyle P, Melchi CF, "Meta-analysis of risk factors for cutaneous melanoma: I. Common and atypical nevi". Eur J Cancer.,41(1):28-44, 2005.
[17] M Y Xiong, M S Rabkin, M W Piepkorn, R L Barnhill, Z Argenyi, L Erickson, J Guitart, L Lowe , C R Shea, M J Trotter, R A Lew, M A Weinstock, "Diameter of dysplastic nevi is amore robust biomarker of increased melanoma risk than degree of histologic dysplasia: a case-control study". J Am Acad Dermatol.,71(6):1257-1258.e4, 2014.
[18] Rastgoo, M., Garcia, R., Morel, O., and Marzani, "Automatic differentiation of melanoma from dysplastic nevi". Computerized Medical Imaging and Graphics, 43, 44-52, 2015.
[19] Barata C, Marques JS, Rozeira J., "The role of keypoint sampling on the classification of melanomas in dermoscopy images using bag-of-features". In: Pattern recognition and image analysis. Springer, p. 715âĂŞ23, 2013.
[20] Quellec G, Cazuguel G, Cochener B, Lamard M, "Multiple instance learning for medical image and video analysisâĂİ, IEEE Rev Biomed Eng 10, pp:213âĂŞ234,

2017.
[21] Astorino, A., Fuduli, A., Veltri, P., and Vocaturo, E.,"On a recent algorithm for multiple instance learning. Preliminary applications in image classification". In IEEE International Conference on Bioinformatics and Biomedicine (BIBM) , pp. 1615-1619, 2017.
[22] Astorino, A., Fuduli, A., Gaudioso, M., and Vocaturo, E., "A multiple instance learning algorithm for color images classification". In Proceedings of the 22nd International Database Engineering and Applications Symposium (pp. 262-266). ACM, 2018.
[23] Astorino, A., Fuduli, A., Veltri, P., and Vocaturo, E. (2019). Melanoma detection by means of Multiple Instance Learning. Interdisciplinary Sciences: Computational Life Sciences, 1-8, 2019.
[24] Litjens, G. and Kooi, T. and Bejnordi, B. E. and Setio, A. A. A. and Ciompi, F. and Ghafoorian, M. and Van Der Laak, J. A. and Van Ginneken, B. and Sánchez, C. I., "A survey on deep learning in medical image analysisâĂİ. Med. Image Anal. 42, 60âĂŞ88, 2017.
[25] Weese, J. and Lorenz, C., 2016. "Four challenges in medical image analysis from an industrial perspectiveâĂİ. Med. Image Anal. 33, 44âĂŞ49, 2016.
[26] de Bruijne, M. , âĂIJMachine learning approaches in medical image analysis: from detection to diagnosisâĂİ. Med. Image Anal. 33, 94âĂŞ97, 2016.
[27] Amores J., "Multiple instance classification: review, taxonomy and comparative studyâĂİ. Artificial Intelligence 201:81âĂŞ105, 2013.
[28] Carbonneau M.A., Cheplygina V., Granger E., Gagnon G., "Multiple instance learning: a survey of problem characteristics and applicationsâĂİ. Pattern Recogn 77:329âĂŞ353, 2018.
[29] de Oliveira, W., "Proximal bundle methods for nonsmooth DC programming", Journal of Global Optimization, pp.1-41, 2019.
[30] Gaudioso, M., Giallombardo, G., Miglionico, G., "Minimizing piecewise-concave functions over polyhedra", Mathematics of Operations Research, vol.43, n. 2, pp. 580-59, 2017.
[31] Gaudioso, M., Giallombardo, G., Miglionico, G., and Bagirov Adil M, "Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations", Journal of Global Optimization, v. 71, n. 1, pp. 37-55, 2018.
[32] Joki, K., Bagirov, Adil M., Karmitsa, N. and Makela, Marko M,"A proximal bundle method for nonsmooth DC optimization utilizing nonconvex cutting planes", Journal of Global Optimization, v.68, n. 3, pp. 501-535, 2017.
[33] Joki K., Bagirov A.M., Karmitsa N., MakelÃď M.M. and Taheri S., "Double bundle method for finding Clarke stationary points in nonsmooth DC programming", SIAM Journal on Optimization, v. 28, n. 2, pp. 1892-1919, 2018.
[34] Schumacher M., Holländer N., Sauerbrei W.,"Resampling and cross-validation techniques: a tool to reduce bias caused by model building", Statistics in medicine, v.16, n. 24, pp. 2813-2827, Wiley Online Library, 1997.
[35] MendonÃğa T., Ferreira P.M., Marques J.S., Marcal A.R.S., Rozeira J., "Ph2 âĂŤa dermoscopic image database for research and benchmarking". In: 35th annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp 5437âĂŞ5440, 2013.
[36] Dietterich T.G., Lathrop R.H., Lozano-PÃlrez T., "Solving the multiple instance problem with axis-parallel rectanglesâĂİ, Artificial Intelligence 89(1âĂŞ2): pp. 31âĂŞ71, 1997.
[37] Andrews S., Tsochantaridis I., Hofmann T., "Support vector machines for multiple-instance learningâĂİ, Advances in neural information processing systems, pp. 577-584, 2003.
[38] Carson, C. and Thomas, M. and Belongie, S. and Hellerstein, J. M. and Malik, J., " Blobworld: A system for region-based image indexing and retrievalâĂİ, International conference on advances in visual information systems, pp. 509-517, 1999.
[39] Sanghera R. and Grewal P. S., "Dermatological symptom assessmentâĂİ, in Patient Assessment in Clinical Pharmacy, p. 133âĂŞ154, Springer, 2019.
[40] Barata C, Ruela M, Francisco M, MendoncÂÿ a T, Marques J. Two systems for the detection of melanomas in dermoscopy images using texture and color features. IEEE Syst J 8(3), pp: 965âĂŞ79, 2014.
[41] Astorino, A., Fuduli, A., and Gaudioso, M., "A Lagrangian relaxation approach for binary multiple instance classification". IEEE transactions on neural networks and learning systems, 2019.
[42] A. Astorino, A. Fuduli, G. Giallombardo, G. Miglionico, " SVM-Based Multiple Instance Classification via DC Optimization", Algorithms, v. 12, n.12, pp. 249, 2019.
[43] Vapnik V., "The nature of the statistical learning theoryâĂİ, Springer, New York 1995.
[44] E. Vocaturo, and E. Zumpano, and P. Veltri, "On the Usefulness of Pre-Processing Step in Melanoma Detection Using Multiple Instance LearningâĂİ, International Conference on Flexible Query Answering Systems, Springer, pp. 374-382, 2019.
[45] E. Vocaturo, and E. Zumpano, and P. Veltri, "Image preprocessing in computer vision systems for melanoma detection,âĂİ IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 2117-24, 2018.
[46] E. Vocaturo, and E. Zumpano, and P. Veltri, "Features for Melanoma Lesions Characterization in Computer Vision SystemsâĂIJ, 9th International Conference on Information, Intelligence, Systems and Applications, (IISA) 2018, Zakynthos, Greece, July 23-25, pp. 1–8, 2018.
[47] Vocaturo E., Caroprese L., Zumpano E., "Features for Melanoma Lesions: Extraction and Classification", WI âĂŹ19 Companion, October 14âĂŞ17, 2019, Thessaloniki, Greece https://doi.org/10.1145/3358695.3360898, in Press, 2019.
[48] E. Vocaturo, E. Zumpano, P. Veltri, "On discovering relevant features for tongue colored image analysis", IDEAS 2019: 12:1-12:8
[49] A. Astorino, A. Fuduli, M. Gaudioso, and E. Vocaturo, "Multiple Instance Learning Algorithm for Medical Image ClassificationâĂİ, Proceedings of the 27th Italian Symposium on Advanced Database (SEDB), 2019.
[50] A. Fuduli, P. Veltri, E. Vocaturo, E. Zumpano, "Melanoma detection using color and texture features in computer vision systems", Advances in Science, Technology and Engineering Systems Journal, vol. 4, no. 5, pp. 16-22 (2019).
[51] E. Zumpano, P. Iaquinta, L. Caroprese, G.L. Cascini, F. Dattola, P. Franco, M. Iusi, P. Veltri, E. Vocaturo, "SIMPATICO 3D: A Medical Information System for Diagnostic Procedures", BIBM pp. 2125-2128, 2018.
[52] E. Zumpano, P. Iaquinta, F. Dattola, L. Caroprese, G. Tradigo, P. Veltri, E. Vocaturo, "SIMPATICO 3D Mobile for Diagnostic Procedures", IIWAS 2019, in press.
[53] E. Vocaturo, P. Veltri, "On the use of Networks in Biomedicine", FNC/MobiSPC 2017: 498-503.
[54] Vocaturo E., and Zumpano E., and Veltri P. "On discovering relevant features for tongue colored image analysis", Proceedings of the 23rd International Database Applications & Engineering Symposium, IDEAS, Athens, Greece, June 10-12, 2019, pp. 1-8.

QoC enhanced semantic IoT model

Hela Zorgati
Higher Institute of Computer Science and Multimedia of Sfax/University of Sfax
Sfax, Tunisia
hela.zorgati@gmail.com

Raoudha Ben Djemaa
Higher Institute of Computer Science and Communication Techniques of H. Sousse/University of Sousse
Sousse, Tunisia
raoudha.bendjemaa@isimsf.rnu.tn

Ikram Amous Ben Amor
National School of Electronics and Telecommunications of Sfax/University of Sfax
Sfax, Tunisia
ikram.amous@isecs.rnu.tn

Florence Sedes
Toulouse Institute of Computer Science Research/Paul Sabatier University
Toulouse, France
florence.sedes@irit.fr

ABSTRACT

The miniaturization of computers, coupled with a constant increase in computing power, led to the emergence of new sources of context information. We are facing a new paradigm, the Internet of Things (IoT). Today, this latter improves the quality of life in multiple areas. However, the heterogeneity of objects used in such environments makes their interoperability difficult. In addition, the observations produced by context providers (connected objects) are generated with different vocabularies and data formats. This heterogeneity of technologies in the IoT world makes it necessary to adopt generic solutions. Therefore, it is important to transform the raw data from these context producers into knowledge and information based on ontologies. The use of ontologies solves the challenges of heterogeneity and interoperability of IoT systems. In this paper, we propose a semantic IoT model that aims to overcome the semantic interoperability challenges introduced by the variety of objects potentially used in IoT systems. Furthermore, we enhanced this ontology with quality of context meta-data. These meta-data helps in dealing with imperfection and inconsistency of the collected IoT data.

CCS CONCEPTS

• **Theory of computation** → **Semantics and reasoning**; • **Computing methodologies** → **Model development and analysis**; • **Human-centered computing** → *Ubiquitous and mobile computing*; • **Computer systems organization** → *Sensors and actuators.*

KEYWORDS

Internet of Things, Semantic model, QoC meta-data, Ontology reuse

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410610

ACM Reference Format:
Hela Zorgati, Raoudha Ben Djemaa, Ikram Amous Ben Amor, and Florence Sedes. 2020. QoC enhanced semantic IoT model. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3410566.3410610

1 INTRODUCTION

Recently, connected objects have experienced a real integration in our everyday lives. We are facing a new paradigm: the Internet of Things (IoT). It has grown incredibly, including more technologies. Besides, the integration of various sensor and actuator systems remotely controllable makes the realization of a new range of intelligent application possible. These applications combine information from multiple sources to provide new services. Due to the number of sources of information available, the services provided are increasingly specialized and adapted to the user's needs. However, these sources are very heterogeneous with different data and services representations. In addition, the vast and complex IoT ecosystem of technologies, protocols and standards makes difficult the deployment of applications integrating different technologies or different sources of data. Thus, the concretization of the IoT vision raises considerable scientific challenges. A first major challenge lies in interoperability. To share information between various IoT system's actors, all actors must adopt the same data and services models so that they understand each other. Unless using the same information model, the applications will have to be rewritten so that they take into account the nomenclature adopted by the other. In the context of a shared platform, a data and service description model allows the existing actors to easily utilize the sensors and actuators of others. In addition, this allows a new actor to be integrated quickly to the platform since it has only to use the predefined model to interact with other actors. In fact, to realize systems interoperability, it is necessary to use different technologies seamlessly and homogeneously by integrating heterogeneous and incompatible technologies. A second step consists in using different technologies in a harmonized way and allowing systems to automatically understand the different data produced by these heterogeneous components with disparate data formalisms. To solve the problems related to the physical heterogeneity of objects, the service oriented architecture (SOA) is used to connect physical objects to the Internet and expose their resources as IoT services. To overcome

syntactic heterogeneity issues in the descriptions of the IoT services, Semantic Web technologies are used. The most considered Semantic Web technology are ontologies [17]. Indeed, many semantic IoT models that uses ontologies to model IoT systems were proposed in the literature. The analysis of the existing research works on the IoT models disclosed some important limitations. First, a unified IoT model that is shared over different IoT actors is needed. Second, an important aspect of IoT systems that must be covered by IoT models is context-awareness. In fact, context-awareness allows intelligent adaptation of the IoT applications so that they can perform their tasks efficiently, proactively and autonomously. Context data is usually perceived in its raw form. Semantic context description mechanisms are then needed to make it understandable and intelligible by the system. Finally, special attention must therefore be paid to the quality of the information they process. Indeed, information sources can be unstable, prone to malfunctions and provide information that is sometimes incomplete, inconsistent or even contradictory. Quality of context (QoC) modeling allows applications, aware of the QoC level of the information they receive, to make better decisions. In this paper, we propose a QoC enhanced semantic IoT model that addresses the above mentioned issues. Our model promotes ontologies reuse which guarantee more interoperability. In addition, the re-usability of strong concepts of different standard or already recognized ontologies in the field makes a force of the proposed ontology and it is therefore interesting to be based on such a unifying ontology.

In the remainder of this paper, section 2 presents a literature review of IoT semantic models. Section 3 details the proposed semantic IoT model and explains how QoC meta-data are used to enhance this model. Section 4 deals with ontology evaluation. Finally, a conclusion is provided in section 5.

2 SEMANTIC IOT MODELS

Of all the difficulties posed by IoT, the interconnection of objects and their interoperability remain the most important. Indeed, in accordance with the uses for which these objects were designed, the exchange and semantics of information derived from such an interaction vary significantly, contributing to making IoT a rich, but also heterogeneous ecosystem. To contribute to IoT evolution, many works attempt to abstract connected objects and their attributes as services so they can be accessible to all applications without requiring prior knowledge. In addition, special attention was paid to the representation of the corresponding information at a semantic level, which makes it possible to share it and make it interoperable independently of the underlying infrastructures. First, we present an overview of the needs that can be met by ontologies in the field of IoT. Then, we present the ontologies currently available in the field.

2.1 Importance of semantics for IoT

The primary goal of semantics is to define and link resources to simplify their use, discovery, integration and reuse in the widest range of applications. The first contribution of using semantics for IoT lies in the transformation of the raw data collected by IoT objects into knowledge. Semantically annotated, these data can take the status of meaningful information out of the application from which they were collected. This enrichment can be done at different stages of the data life cycle as reported in [7]. By adding meta-data, the data are no longer specific to the application that collected it. Therefore, there is a sense in integrating this information into linked data. Linking a piece of data to remote warehouses increases its value. Finally, semantically annotated, the information generated by the object network can be used by an inference engine.

Since the IoT environment is open, and the IoT service requesters are machines rather than people. This requires a more understandable and interpretable service description by these machines to facilitate the automation of the registration, search, selection and acquisition processes of the required IoT services and cope with the ever-increasing number of them. Seeing that, the classic description of services (WSDL for web services, for example) is not sufficient to carry out these tasks in the long term, some works propose to describe services based on semantic rather than syntactic aspects [4][22]. The use of semantics overcomes these difficulties and encode the IoT services in an unambiguous and understandable form. Thus, the use of semantics for the description of data and the IoT services makes it possible to provide access to these resources via semantic descriptions that is exploitable and comprehensible by machines. In addition, the semantic techniques make it possible to ensure the interoperability and the implementation of treatments and complex reasoning on the knowledge describing the IoT resources. The description of this knowledge is based on the use of ontologies. As defined in [17], an ontology is "an explicit specification of a conceptualization". A conceptualization is an abstract model that represents how people conceive real things in the world and an explicit specification means that the concepts and relationships of an abstract model are given explicit names and definitions. For that reason, we have used ontologies to model the IoT resources and services. Before detailing our proposed model, we will present, in the next section, the most important ontologies used to model the IoT concepts.

2.2 Semantic IoT models: state of the art

Many ontologies have been developed in literature. In the existing ontologies, different categories are distinguished as foundation ontologies, which define very high-level concepts that will be instantiated in other ontologies, domain ontologies that define primordial concepts in a particular domain and specific ontologies that define concepts specific to a particular context or a specific scenario. In the following section, some of the most used ones in IoT domain are detailed.

Semantic Sensor Network (SSN) [9] is an ontology dedicated for modeling sensors (rather than objects in general) and observations on the origin. The SSN ontology consists of modules that provide descriptions of the IoT devices from different perspectives: sensor perspective (what IoT device is sensing), observation perspective (IoT data that is produced by IoT device), system perspective (the IoT application), feature perspective (the IoT data properties), deployment perspective, measurement capability and conditions perspective (the sensor capabilities and the conditions of its observations).Since then, SSN has been reworked in 2017 and separated into an SOSA module (Sensor, Observation, Sample and Actuator) [19] and an extension of this new ontology has been realized to

Table 1: Studied works

Approach	Modeled concepts	Language	Ontologies reuse	QoC meta-data	Domain of application
SSN/SOSA [9][19]	Sensors, system observation features	OWL-LD	No	No	Sensing for manufacturing
One M2M Base ontology [31]	Thing, device, service	OWL-LD	No	No	Domain independent
IoT-Lite [5]	Objects, services	OWL	SSN	No	Environment monitoring
IoT-A [1]	Entity, resource, service	OWL-LD	OWL-S, FOAF	+/-	Objects and resources mapping
SAREF [10]	Appliance, service, energy	RDF	WGS84, OWL-Time	No	Smart appliance reference model
PT-SOA [33]	Physical thing, service	OWL	No	No	Emergency rescue
IoT-O [26]	Sensor, actuator, service, energy	OWL	OWL-S, SSN, SOUPA, PowerOnt	No	Robotics
QoDisco IM [15]	Sensor, actuator, service	OWL	SSN, SAN, OWL-S, SOUPA	Yes	Service discovery

make the connection with the old terminology of SSN and the concepts that have not been included in SOSA.

The oneM2M standards initiative has defined a domain independent ontology, the oneM2M base ontology [31]. This ontology is defined to be able to integrate semantic meta-data to describe the different resources registered in the platform. This ontology is a core domain ontology: a minimal set of concepts is defined to allow subsequent alignment with other concepts standards or specific to different implementations and technologies. Defined concepts are high-level concepts like Object or Service and are related to oneM2M architecture resources.

IoT-Lite [5] is a lightweight core ontology. In IoT-Lite, high-level concepts are defined as actionable objects and services. The model is an instantiation of SSN appropriate for real time sensor discovery. The IoT-Lite ontology has been adopted in different projects like core model because of its simplicity (FiWARE [25] project for example).

The European project Internet of Things Architecture proposed a semantic model for entities, resources and services, the IoT-A ontology [1]. The latter represents all the entities and the relationships involved in an IoT-based system. Services expose features from the resources they have. Two types of resources are taken into consideration; remotely available resources and local resources. Three types of hardware are used: actuators, RFID chips and sensors. The proposed model integrates the notion of meta-data. It is possible to interpret the meta-data of the model as QoC meta-data. In this case, it is possible to represent, using the model, primitive and composite QoC criteria. Each meta-data is specified by a name, a value and a type.

Smart Appliance REFerence [10], SAREF is an ontology dedicated to energy management and services in smart home domains. SAREF combines the generic concepts derived from the semantic annotation of other standards and data models used in the field of intelligent Appliances. The basic concept used in SAREF is that of Device, it represents objects found in the field of smart home. In addition, SAREF uses the notion of function that is presented to the network via a service. Each device offers these functions via associated commands. It is also characterized by an Energy/Power profile to optimize energy efficiency. Since then, SAREF ontology has been extended in [11] to be applied to other areas of IoT such as connected buildings, etc.

In [33], the authors define a PT-SOA ontology for physical things (PTs) and services in the cyber-physical domain. The authors model the PTs as providers/recipients of services. They proposed an ontology that has four classes: 'physical profile' which maintains PT physical characteristics, PT selection, and service invocation; 'operation profile' that specifies the properties that affect the operation and services of the PT; 'scheduled service' used to schedule multiple requests to PT provided services and 'context' to define the dynamic state of the PT.

IoT-Ontology described in [26], is a unifying ontology that merges together many concepts defined in recognized ontologies. The design of IoT-O respects the NeON methodology [30]. Sensor concepts are based on SSN definitions. Actuator concepts are based on SAN and SSN concepts. Service management is based in particular on MSM and WSMO and also integrates elements of hRESTs. The integration of energy is carried out using the concepts defined in PowerOnt[6].

The information model of QoDisco [15] uses an ontology-based vocabulary of concepts related to the IoT resources and services, relationships among them, and QoC-related information. The proposed model uses SSN ontology to model sensors and SAN ontology to model actuators. Additionally, the model incorporates a part of the SOUPA ontology [8] to describe location-related concepts. The QoDisco information model also uses the OWL-S ontology [21] to model Web services that are exposed by resources. The model, also, defines QoC-related concepts. The QoC Criterion concept is used

to represent a QoC parameter associated with context data and the QoC Indicator concept expresses the QoC level of an observation.

Based on the findings from the analysis of the different IoT description approaches, we can conclude that providing a semantic web-based model for the IoT systems allows to unblock the analysis skills of this system based on the existing vocabularies and using the power of semantic reasoners. In fact, semantic models make it possible to store information in the form of a semantic knowledge base as well as to make an inference of knowledge if necessary. Moreover, using a comprehensive and complete IoT model can facilitate the IoT service discovery and selection tasks, and can promote IoT systems interoperability [34].

We notice, from Table 1, that studied works provide interesting solutions for the IoT domain modeling. Although, some proposed semantic IoT models cover the most important concepts of the IoT domain [26] [15] , the QoC meta-data were not well integrated with these models. Nevertheless, a question remains as to the quality management of the information received by the applications. We notice, in these works, the lack of definition of the QoC used to assess the quality of information produced by the information sources. We find also, that some works [33] do not reuse the existing ontologies, even though the best practice of ontology development consider reusing existing ontologies [27]. This leads to the redefinition of the already existing concepts, which raises alignment efforts in order to be integrated with the existing systems.

To address these issues, we propose an IoT semantic model that reuses the strong concepts of existing ontologies. Furthermore the proposed model is enhanced with data quality metrics. Thus, the proposed IoT model can facilitate the IoT service discovery and selection tasks, and can promote IoT systems interoperability.

3 A QOC ENHANCED SEMANTIC IOT MODEL

In this section, we present the main concepts of our proposed semantic IoT model. Then, we focus on enhancing this semantic model with QoC meta-data.

3.1 Top-level ontology concepts

Fig. 1 represents the top-level concepts of the proposed ontology. This ontology was conducted considering the previously studied literature. In our ontology, we consider reusing the existing ontologies. Therefore, sensors and actuators concepts are based on the Sensor, Observation, Sample, and Actuator (SOSA) ontology. In fact, SOSA ontology provides a formal and a lightweight general-purpose specification for modeling the interaction between the entities involved in the acts of observation, actuation, and sampling [19]. As SOSA is the result of the rethinking of SSN that integrates actuator definition, and as SSN ontology is widely accepted for sensor definitions, our choice was particularly oriented towards SOSA.

As we have reported previously, to facilitate the integration of things and to solve problems related to the physical heterogeneity of objects, the widely used approach is to abstract these latter as services following SOA approach. In our ontology, service management is based in particular on OWL-S. In fact, inspired by the IoT-A project, we used a modified version of the OWL-S. Due to the restricted IoT objects' capabilities, the Profile-Process-Grounding design pattern is replaced by the Profile-Model-Grounding design pattern to avoid the complexity related to the Process. As in [12], the service model represents the resource functionalities in terms of the input, output, precondition, effect terms. Defined concepts in the QU (Quantity Kinds and Units) ontologies [13] are used to define the input and output types. We have related the Service concept to the Resource concept by the expose property to model the fact that a service provides functionalities to provide information about the entities they are associated with or to manipulate the physical properties of the related entities or their surrounding environment.

An important aspect in IoT is the location of connected objects. In fact, the commonly used queries in IoT are location-based queries. Therefore, we have used the Location concept to model the geographical location of an IoT object. This concept is related to the Resource concept through the hasLocation property. The Integration of location is carried out thanks to the concepts defined in the WGS84 ontology [2]. In addition, these concepts are mapped to GeoNames place names to enable the semantic reasoning needed by the relative positioning.

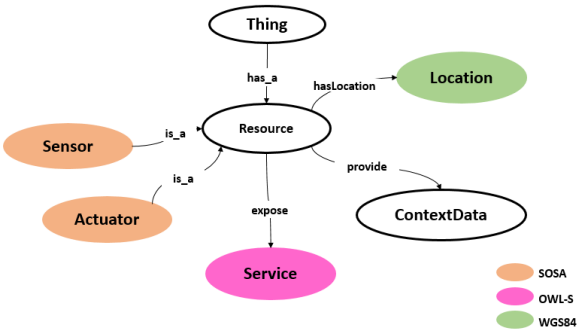


Figure 1: Top-level ontology concepts.

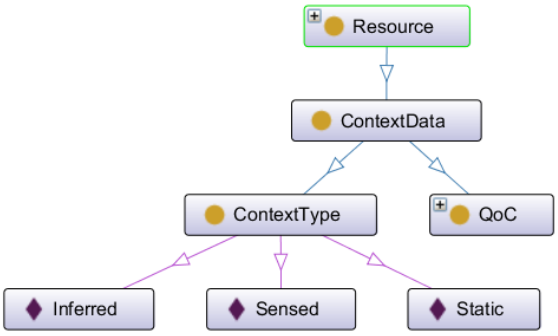


Figure 2: Context Data concept.

Finally, to model the fact that a resource of an IoT object is used to provide information about the surrounding context, we have used the object property provide which links the Resource concept to ContextData concept. Fig. 2 represents the ContextData concept

subclasses. Each context data delivered by the IoT objects has a context type represented by the ContextType concept. The latter represents the acquisition mode of the context data. We distinguish three context types: static, sensed and inferred context.

In the next section we will detail the sub-concepts of the QoC concept. Nevertheless, before that, we will discuss the notion of quality of context (QoC) in the IoT domain.

3.2 The QoC relevance in the IoT domain

IoT offers new sources of context information where these pieces of information are no longer provided by people, but directly from everyday objects that surround us. The information they produce is, therefore, numerous and heterogeneous. These are most often raw pieces of information from sensor measurements. Connected objects can be mobile and therefore provide information sporadically. This heterogeneous and dynamic nature of IoT can adversely affect the accuracy and consistency of the context data provided by the IoT objects. Nevertheless, there is still a problem with managing the quality of the context information. The question that arises is how to know if an application suggests a wrong indication after receiving inaccurate information. To answer this question, the quality of the context information must be supported as early as the IoT data modeling stage.

The Quality of Context (QoC) extends the notion of Quality of Information (QoI) for context information. According to [14], the Quality of Context is defined as follows: "We define QoC as the set of parameters useful for expressing properties and quality requirements on context data". Then, the QoC information is metadata associated with the context information used to calculate a quality level of the contextual information.The omission of QoC can lead to poor decision-making or irrelevant responses for context-sensitive applications. The QoC management, therefore, appears as a non-functional need that must be taken into account to ensure the success of the new generation of the IoT applications.For all these reasons, we decided to incorporate QoC criteria into our proposed ontology. The following section details the QoC criteria we selected for our model.

3.3 QoC sub-concepts

There is a lack of consensus regarding the definition of a common list of QoC criteria to be used to qualify context information. In fact, the QoC criteria that are reported in the literature [3] are heterogeneous and form non-exhaustive lists that manipulate different notions. We have conducted, then, a study to determine the QoC criteria relevant to qualifying the quality of context information. An important work in this field is the survey in [20]. In this work, the authors divided data quality concepts in IoT into two categories: generic data quality dimensions (Accuracy, Confidence, Completeness, Data volume, Timeliness, Ease of access, Access security, Interpretability) and domain-specific data quality dimensions (Duplicates, Availability). Despite providing an interesting review of data quality dimensions, this work does not provide a solution to integrate this quality dimensions in a unified IoT description model. In the following, we provide the QoC criteria that we have integrated to our model. Fig. 3 depicts the different QoC sub-concepts.

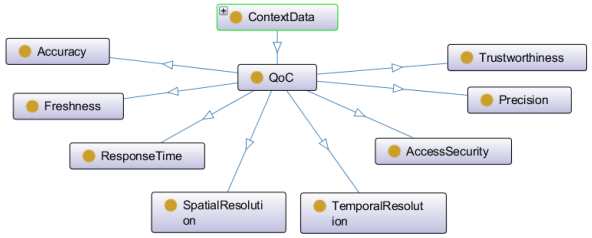


Figure 3: QoC sub-concepts.

The *TemporalResolution* concept represents the time elapsed between two measurement productions.

The *SpacialResolution* concept represents the geographic precision of context information. For example, the GPS coordinates are more precise than street name.

The *Precision* concept represents the granularity with which the context information describes a real world situation. We have borrowed this definition from [23].

The *Accuracy* concept represents the probability that the context information correctly describes the real-world situation it represents. This QoC criterion is also used under the name 'probability of correctness'.

The *ResponseTime* concept represents the time elapsed between the request reception and the response availability.

The *Freshness* concept represents the time elapsed between the production of the context information and the reception of this context information by an application. In other words, this concept represents the age of the context information.

For reasons of completeness, to cover an important aspect of the IoT system, namely security, we have introduced the criteria that reflect the security level of the context information. These criteria are described below.

The *TrustWorthiness* concept represents the level of confidence that the application grant to the different sources of information that they use.

The *AccessSecurity* concept represents the probability with which the context information has been transmitted from source to applications over secure networks. The evaluation of this criterion is based on configuration files.

The adoption of an exhaustive list of context quality criteria with precise definitions facilitates the use and understanding of these criteria. Indeed, the integration of context quality criteria like generic classes, as in [15], leads to confusion regarding the names and definitions of the criteria used when instantiating the model. For example, as reported by the authors of [3], the precision criteria are used in literature as measurement of both accuracy, precision and probability.Taking these reasons into account, we believe that the QoC criteria must be clearly defined in an exhaustive list.

4 ONTOLOGY EVALUATION

To construct our ontology, we followed two steps. First, we identified the relevant concepts by analysing proposals from literature. Second, we applied the design principles, presented in [16], which are the objective criteria for developing and evaluating ontology

Figure 4: An extract from a hazardous gas detection service description.

designs. They include clarity, minimal encoding bias, extendibility, coherence, and minimal ontological commitments. To design our ontology, we used the Protègè editor, which is the most used one. Protègè is freely available for developing and managing terminologies, ontologies and knowledge bases. Once designed, we have evaluated our ontology in terms of qualitative and quantitative aspects. The qualitative evaluation deals with the conceptual coherence while the quantitative evaluation deals with structural criteria [24].To improve the qualitative evaluation of our proposed ontology, we refer to the criteria defined in [29]. These criteria are:

- Duplication: checks if the elements that can be deducted need not to be explicitly mentioned.
- Disjunction: checks if the class is the conjunction of disjoint classes.
- Consistency: checks if the definition of a class does not lead to a contradiction.

The validation of ontologies structure has led to numerous studies and applications. We can cite the reasoners PELLET [28] and Fact++ [32] among the most used because of their presence as an extension module in Protégé. To validate the proposed ontology, we used the Fact++ [32] reasoner. Therefore, we have applied the reasoner in order to obtain a coherent and valid IoT semantic model. For quantitative evaluation, in order to realize performance tests for our proposed ontology, we created three datasets containing 100, 1.000 and 10.000 sensors and services descriptions each. Fig. 4 present an extract from the description of a hazardous gas detection sensor based on the proposed model. Then, we performed a common query written in DL-Query [18] (Fig. 5) that asks for the services that provide the hazardous gas rate in a particular area over different datasets. Fig. 6 present the response time of the proposed model for this query. We have found that the response time for the query is acceptable for different datasets. In fact, it is around 100 milliseconds for the dataset that contains 10000 individuals.

```
Service and ServiceName value hazardousGasesDetectionService
and has_Accuracy min 0.9 and hasLocation value "Gabes"
```

Figure 5: Query used in the experiments.

Figure 6: Response time of the proposed model to the test query.

5 CONCLUSIONS

To share information between the various actors of an IoT system, all actors must adopt the same data and services model, so that these actors understand each other. In this context, we have proposed an ontology, which serves as a unifying IoT semantic model. In the proposed ontology, we have reused the concepts of the ontologies already existing in the field to contribute to a better level of interoperability of the IoT systems. In addition, we have enriched our model with QoC meta-data. These QoC meta-data are important to integrate since they reflect the ambiguous, imprecise and erroneous nature of the context data provided by the connected objects. Therefore, by being aware of the level of QoC, context-aware applications can make better decisions.

As a future work, we plan to improve the evaluation of the proposed ontology. As an improvement, we will populate the ontology with real individuals from available datasets. Moreover, we intend to use the proposed ontology as a reference model in order to search and discover the IoT services.

6 ACKNOWLEDGMENTS

This work was funded by the “PHC Utique” program of the French Ministry of Foreign Affairs and Ministry of higher education and research and the Tunisian Ministry of higher education and scientific research in the CMCU project number 18G1431.

REFERENCES

[1] M. Bauer and J. W. Walewski. The iot architectural reference model as enabler. In *Enabling Things to Talk*, pages 17–25. Springer, Berlin, Heidelberg, 2013.
[2] C. Becker and C. Bizer. Exploring the geospatial semantic web with dbpedia mobile. *Journal of Web Semantics*, 7(4):278–286, 2009.
[3] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini. A survey of context data distribution for mobile ubiquitous systems. *ACM computing surveys (CSUR)*, 44(4):1–45, 2012.
[4] R. Ben Djemaa, H. Nabli, and I. Amous Ben Amor. Enhanced semantic similarity measure based on two-level retrieval model. *Concurrency and Computation: Practice and Experience*, 31(15):e5135, 2019.
[5] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor. Iot-lite: a lightweight semantic model for the internet of things. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 90–97. IEEE, 2016.
[6] D. Bonino, F. Corno, and L. De Russis. Poweront: An ontology-based approach for power consumption estimation in smart homes. In *International Internet of Things Summit*, pages 3–8. Springer, 2014.
[7] J.-P. Calbimonte, O. Corcho, and A. J. Gray. Enabling ontology-based access to streaming data sources. In *International Semantic Web Conference*, pages 96–111.

Springer, 2010.
[8] H. Chen, T. Finin, and A. Joshi. The soupa ontology for pervasive computing. whitestein series in software agent technologies, 2005.
[9] M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, et al. The ssn ontology of the w3c semantic sensor network incubator group. *Journal of Web Semantics*, 17:25–32, 2012.
[10] L. Daniele, F. den Hartog, and J. Roes. Created in close interaction with the industry: the smart appliances reference (saref) ontology. In *International Workshop Formal Ontologies Meet Industries*, pages 100–112. Springer, 2015.
[11] L. Daniele, M. Solanki, F. den Hartog, and J. Roes. Interoperability for smart appliances in the iot world. In *International Semantic Web Conference*, pages 21–29. Springer, 2016.
[12] S. De, T. Elsaleh, P. Barnaghi, and S. Meissner. An internet of things platform for real-world and digital objects. *Scalable Computing: Practice and Experience*, 13(1):45–58, 2012.
[13] H. de Koning, N. Rouquette, R. Burkhart, H. Espinoza, and L. Lefort. Library for quantity kinds and units: schema, based on qudv model omg sysml (tm), version 1.2. Technical report, CSIRO, Tech. Rep., 2011, retrieved: 04, 2016.[Online]. Available: http://www …, 2009.
[14] M. Fanelli, L. Foschini, A. Corradi, and A. Boukerche. Qoc-based context data caching for disaster area scenarios. In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2011.
[15] P. Gomes, E. Cavalcante, T. Batista, C. Taconet, D. Conan, S. Chabridon, F. C. Delicato, and P. F. Pires. A semantic-based discovery service for the internet of things. *Journal of Internet Services and Applications*, 10(1):1–14, 2019.
[16] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5-6):907–928, 1995.
[17] T. R. Gruber et al. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–221, 1993.
[18] M. Horridge and N. Drummond. Dlquery, 2008.
[19] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois. Sosa: A lightweight ontology for sensors, observations, samples, and actuators. *Journal of Web Semantics*, 56:1–10, 2019.
[20] A. Karkouch, H. Mousannif, H. Al Moatassime, and T. Noel. Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications*, 73:57–81, 2016.
[21] D. Martin, M. Paolucci, S. McIlraith, M. Burstein, D. McDermott, D. McGuinness, B. Parsia, T. Payne, M. Sabou, M. Solanki, et al. Bringing semantics to web services: The owl-s approach. In *International Workshop on Semantic Web Services and Web Process Composition*, pages 26–42. Springer, 2004.
[22] H. Nabli, R. B. Djemaa, and I. A. B. Amor. Linked usdl extension for cloud services description. In *International Conference on Web Engineering*, pages 359–373. Springer, 2019.
[23] R. Neisse. *Trust and privacy management support for context-aware service platforms*. University of Twente, Enschede, Netherlands, 2012.
[24] C. Pradel, N. Hernandez, M. Kamel, and B. Rothenburger. Une ontologie du cinéma pour évaluer les applications du web sémantique. In *Atelier Ontologies et Jeux de Données pour évaluer le web sémantique, IC'2012*, 2012.
[25] F. Ramparany, F. G. Marquez, J. Soriano, and T. Elsaleh. Handling smart environment devices, data and services at the semantic level with the fiware core platform. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 14–20. IEEE, 2014.
[26] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil. Iot-o, a core-domain iot ontology to represent connected devices networks. In *European Knowledge Acquisition Workshop*, pages 561–576. Springer, 2016.
[27] E. Simperl. Reusing ontologies on the semantic web: A feasibility study. *Data & Knowledge Engineering*, 68(10):905–925, 2009.
[28] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. journal of web semantics, 2007.
[29] S. Staab and R. Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.
[30] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernandez-Lopez. The neon methodology framework: A scenario-based methodology for ontology development. *Applied ontology*, 10(2):107–145, 2015.
[31] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song. Toward a standardized common m2m service layer platform: Introduction to onem2m. *IEEE Wireless Communications*, 21(3):20–26, 2014.
[32] D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. In *International joint conference on automated reasoning*, pages 292–297. Springer, 2006.
[33] W. Zhu, G. Zhou, I.-L. Yen, and F. Bastani. A pt-soa model for cps/iot services. In *2015 IEEE International Conference on Web Services*, pages 647–654. IEEE, 2015.
[34] H. Zorgati, R. B. Djemaa, and I. A. B. Amor. Service discovery techniques in internet of things: a survey. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 1720–1725. IEEE, 2019.

The Web: a hacker's heaven and an on-line system

Bipin C. Desai*
BipinC.Desai@concordia.ca
Concordia University
Montreal, Canada

Arlin L. Kipling
kipling@cse.concordia.ca
Concordia University
Montreal, Canada

Reethu Navale
r_navale@encs.concordia.ca
Concordia University
Montreal, Canada

Jainhu Zhu
fishsb19@gmail.com
Concordia University
Montreal, Canada

ABSTRACT

The internet was supposed to be an interconnection of independent distributed computer and information systems; the web was formally introduced in 1994 at the first conference now known as WWW1 in Geneva, It was supposed to make easier access to a trove of decentralized, independently owned information, The web has made it possible for billions of users to access the internet and its resources. As with any project, whether software or not, unless it is thoroughly thought out, the final outcome has bugs, commissions, omissions, vulnerabilities, and shortfalls. The web has made it possible for a small number of corporations to amass huge quantities of private information and mine them for profit. In this survey paper, we have shown how some of these shortfalls of the web and have impacted CrsMgr, an online course management system and what has been attempted to address these issues.

CCS CONCEPTS

• **Security and privacy → Firewalls**; **Privacy protections**; • **Networks → Middle boxes / network appliances**; • **General and reference** → *Experimentation.*

KEYWORDS

Privacy, security, online web applications, hacking, web security issues

ACM Reference Format:
Bipin C. Desai, Arlin L. Kipling, Reethu Navale, and Jainhu Zhu. 2020. The Web: a hacker's heaven and an on-line system. In *24th International Database Engineering & Applications Symposium(IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3410566.3410589

*Corresponding Author

IDEAS 2020, August 12–14, 2020, Seoul,Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410589

1 INTRODUCTION

In the early days of computing, users had to manually look into several documents to access the information stored in various files on one or more systems. If the documents were on different systems, s/he had to log in to each. A Conseil Européen pour la Recherche Nucléaire (CERN) programmer using the work done by others in hypertext, based on the ideas expressed by Bush in a 1946, article introduced the HTTP protocol and HTML markup language. The later was a stripped-down version of the SGML Web in the late 1980s Hypertext concept allowed related information from various sources to be accessible by linking them. Soon graphical web browsers such as Mosaic, developed at the National Center for Supercomputing Applications at the University of Illinois, gained popularity. Some statistics from January 1994 show the web growth of over 2 million users (150,000 new users a month). Many users used the Mosaic browser as it had an additional feature that images could be added to web pages unlike text-based browsers initially developed at the CERN. In May 1994, the First World Wide Web Conference was held at CERN. Until early 1990s internet and then the early web using a text browser were mainly used by academics and technology companies who were already connected. However, from 1995 many corporations started investing in the web. The Web was no longer restricted to computers. After many years, users can use web browsers from their mobiles and tablets. Data is a treasure in the computing world and the security of this data is very important. Some of the web technologies pay lip-service to privacy and security.Web technology, which allows third party links and contents, allows hackers to access the data which were supposed to be private and non-accessible. These have made the web a hacker's heaven. CrsMgr, developed during these early days of the web had to contend with this hostile environment and required updates discussed here.

2 THE WEB

In 1945, Bush a computer scientist wrote an article "As We May Think" which was described as visionary and influential, anticipating many aspects of the information society. It was first published in The Atlantic in July 1945 [3], and an edited version was republished in September 1945. The article was about a photo-electrical-mechanical device called a Memex, for memory extension, which could make and follow links between documents on microfiche [32][28]. In the 1960s, a computer collaboration system NLS system was the first to employ the practical use of hypertext links[8] and

other modern computing concepts [29]. In 1963, the word Hypertext was coined and a hypertext editing system was developed[32].

The late 1980s was the crucial period in the birth of Web when a programmer working at CERN. developed ENQUIRE, a simple hypertext program that was the predecessor of the World Wide Web. The first web server was developed in the early 1990s using the HyperText Transfer Protocol (HTTP) and the HyperText Markup Language (HTML). Mosaic was the first graphical browser and it was followed by others. Currently the web browsers are part of mobile and hand held devices. The graphical browser and the web gave rise to a number of corporations in the late 1990s and early 2000s which now dominate the use of internet and the web.

3 SECURITY ISSUES

As data is shared and since there is hardly any traditional social or legal protection, the threats of misuse have intensified. The fact that user's data is appropriated for other purposes and mined is in fact a robbery yet there is hardly any easy remedy. It is increasingly recognized that personal data has value, and the organization which collects the data could access this information for their gain. Hackers try to do so using their discovery of weakness in various protocols.

Course Manager (CrsMgr), a web application developed in the early days of the web, is a system to manage the course material, students work, manage groups, provide a medium for online assessments and online quizzes. The newly discovered security issues were studied and the mechanism to prevent data from being hacked were put in place[33]. Security problems such as SQL injection and Cross-Site Scripting (XSS) could allow a hacker, who may be a user or not, to directly query the database by passing a SQL script in the input if there is no mechanism to verify input from users. Hackers could easily access the database and modify the data or collect sensitive information or even delete the information from the database. Since there was no validation to the user's input, hackers could also inject JavaScript with normal textual information into the text input area. When an unsuspecting user visits the corresponding web page, which contains the application's normal textual information along with the hacking script, the normal information will be rendered as usual by the browser. However, the script will be executed by the browser like any other script. Hackers can use appropriate scripts to steal a victim's cookie. Once the hacker has a victim's cookie some plugin tools such as "EditThisCookie" [4] can be used to find details like a session ID and access or modify information.

3.1 Cookies

Cookie also called HTTP Cookie, Web Cookie, Internet Cookie, or Browser Cookie is a small text file that is stored on the user's computer browser directory or program data subfolders [1]. A cookies are created when a user uses the browser to visit a website that uses cookies to keep track of their movements within the site. A cookie helps in resuming a web session, remembering the previous login details, theme selection, preferences, and other customization functions. They can also be used to remember pieces of information that the user previously entered into form fields[12].

There are two types of cookies: session cookies and persistent cookies [1]. Session cookies are created temporarily in the browser's sub-folder while visiting a website. The session cookies get deleted once the session is deleted. Persistent cookie files stay in the browser's sub-folder and get reactivated on a re-visit of the corresponding website. A cookie has a duration period set within the cookie's file and controlled by the user using the browsers preference menu.

Cookies contain random alphanumeric text characters and vary from site to site. Information in the cookies is coded and is not easily undecipherable to anyone accessing the cookie folder. One of the ways of making the web secure is using a secure cookie which can only be transmitted over an encrypted connection (i.e., HTTPS) and over non-secure connections (i.e., HTTP). Another way to secure the web is to use an Http-only cookie that cannot be accessed by client-side APIs, such as JavaScript or any cross-site scripting. However, the cookie remains vulnerable to cross-site tracing (XST) and cross-site request forgery (XSRF) attacks. In 2016 Google introduced a new kind of cookie that will send cookies with requests originated from the same domain/site as the target domain; this will help prevent cross-site request forgery attacks [30]. Supercookies are a big concern and therefore are often blocked by web browsers. If they are not blocked by the web browser then an attacker can easily access the data. For security purposes, it is preferable to block such cookies.

The report on CrsMgr has detailed information on web security[3]. The latest version of CrsMgr uses "HttpOnly" cookies, first implemented in 2002[25]. The majority of web attacks target theft of session cookies. One of the ways to mitigate this issue is to use the HttpOnly flag on a cookie. HttpOnly is an additional flag included in a Set-Cookie HTTP response header. Using the HttpOnly flag when generating a cookie helps mitigate the risk of client-side script accessing the protected cookie. As a result, the browser will not reveal the cookie to a third-party. However, if the browser does not support the HttpOnly flag then this flag will be ignored by the browser thus creating a path for hackers to invade the system and access the data using cross-site scripting (XSS). Along with this cookie additional verification and validation are required to make the web application secure. CrsMgr not only uses "HttpOnly" cookies but also has other preventive measures to secure the application from being hacked. The Session ID stored in a cookie is generated uniquely and encrypted and hashed to make it more secure.

3.2 SQL Injection

Some of the websites need user inputs; such websites are prone to security attacks. Attackers use malicious SQL statements in input data for the website and these SQL statements can either access or modify the data from the database. This code injection technique is called SQL injection and is a security vulnerability in any website or web application[24]. SQL injection attack was considered one of the top web application vulnerabilities from 2007 to 2010. There are four main subclasses of SQL injection: Classic SQL injection, Blind or Inference SQL injection, Database management system-specific SQL injection, and Compound SQL injection. One of the famous SQL injection attacks named "Storm Worm" used the compound SQL injection technique for hacking the data[8, 28]

In general, SQL injection could be categorized as follows:

• **First Order Attack**: the malicious code is injected and executed immediately.
• **Second Order Attack**: the malicious code is saved in persistent storage, considered as credible data, and is executed by another query.
• **Lateral Injection**: the malicious code changes the value of one or more environment variables or implicit functions.

As the SQL injection attacks are evolving so do its solution to stop such attacks. Prevention of SQL injection attacks is difficult but not impossible. Sometimes different techniques need to be used based on the different types of SQL injection. The following methods can be used to protect web application against SQL injection:

• **Check input data**: Do not trust the user input as any user input is a risk of an SQL injection. Always check the data by filtering the keywords, escaping special characters, and type checking.
• **Parameterize the input data**: The query statement is prepared and care is taken before binding the user-supplied data to the parameters of the statement.
• **Usage**: Assign appropriate privilege for the SQL account used for each query.

The report on CrsMgr deals with these security issues: it shows how a web application can identify a malicious SQL injection and can prevent this vulnerability.[3]. This application started out as a PL/1 batch application which evolved into an X-Windows based system[33]. One of the methods used is keyword filtering to filter out likely SQL injection keywords. It checks if there are any special characters in place of text or numeric data. It is difficult to determine if the user's input is an injection or normal input just by using keywords filtration. For example, the user may input a natural language sentence that contains keywords such as: 'and', 'or', 'union', 'select', and so on. Thus, texts could not be handled by simple filtering. Such texts would be handled by injection invalidation. The idea of injection invalidation is, separating the SQL query preparation and parameters binding into two phrases. All parameters and user input are considered as parameters in the source code. CrsMgr only converts SQL query which is written in source code to database query template. Since parameters would never be converted to a query template, SQL injection through parameter hacking is thus protected against and effectively invalidated.

CrsMgr uses parameterized queries to prevent SQL injection attacks: such queries are a means of preparing a SQL query statement so that it can be supplied the parameters for its execution. This method makes it possible for the database to recognize the code and distinguish it from input data. The user input is automatically quoted and the supplied input will change the intent, so this coding style helps mitigate an SQL injection attack [26]. Parameterized queries can be used for any situation where untrusted input appears as data within the query [27]. SQL injection could be inserted in parameters; even for hidden parameters. An earlier version of the CrsMgr was using a direct concatenated SQL query which was a threat as it allowed malicious code to change the web application's SQL statement and access or modify the data. The latest version of CrsMgr removed the direct concatenation of parameters and used parameterized SQL query to fix the security vulnerability. For example, the earlier version of CrsMgr used statement such as:

"SELECT + param1 + FROM XXX WHERE YY = + param2"

It was easy for a hacker to modify and access the data. For example, adding subquery as a parameter to fetch all the details from the XXX table that is:

"SELECT + param1 +FROM XXX WHERE YY = + "'ABC' UNION SELECT * FROM XXX WHERE YY <>'".

If no validation was done for the parameters sent as an input to the query, the query would provide all the details where YY is not NULL and duplicate the value of ABC. As a result, the malicious code would be executed and obtain the results. However, the current version of CrsMgr uses a parameterized query instead of concatenation: for example, using a statement such as :

SELECT ? FROM XXX WHERE YY = ?.

In this case, if a hacker tries to inject malicious code in the statement such as:

SELECT ? FROM XXX WHERE YY = ABC UNION SELECT * FROM XXX WHERE YY <>

The entire second parameter (i.e. along with UNION query) is considered as a value for YY. The SQL will check for this YY value in table XXX and it will not consider it as a subquery. In this case, the select query will not find any value matching the above and returns an empty set. Thus, SQL injection is invalidated and secure the web application from such attacks.

3.3 Cross Site Script (XSS)

Cross-Site Scripting (XSS) is an attacking technique where a malicious script (such as JavaScript or JSP) is injected into the user input area. This malicious script will be executed in the end-user browser and the browser will trust the script as it has come from the trusted website and thus attackers get access to the user's sensitive information such as cookies, session tokens, etc. Microsoft security-engineers introduced the term "cross-site scripting" in January 2000 [6]. Many popular web sites have been affected by XSS such as Twitter, Facebook, MySpace, YouTube, and Orkut [6].

Cross Site Scripting has three main categories namely [13]:

• **Stored XSS**: The malicious code is stored by the attacker on the server-side, typically in persistent storage (database). When the server handles the victim's request, the malicious code is used to construct a web page for the victim.
• **Reflected XSS**: This type of Cross-Site Scripting is executed immediately. The malicious code is sent to the server and carried back to the browser.
• **DOM-based XSS**: The malicious code does not come from the server. Instead, it utilizes the DOM interface to hijack the request

of the victim.

A recent development to avoid hackers and cross-site scripts transmitted via emails is to scan email by the user mail agent for hackers or scripts before it is handed to the outgoing mail system. However, it is always possible to bypass such a mail agent.

To prevent Cross-Site Scripting, the following rules must be observed:

- All untrusted third party or user-supplied content must be escaped before inclusion in the page. Even data from the client's persistence storage may contain malicious script.
- The attribute data in HTML tags, such as font size, width, etc, should be checked.
- Any data going to the scripts (such as JavaScript or CSS) needs to be checked before execution.
- Usage of "HttpOnly" cookies to prevent them from being stolen and blocking third-party cookies.

The CrsMgr report has detailed information on how to deal with cross-site scripting vulnerability [3]. The techniques such as keyword filtering, escaping special characters, and type checking are used[33]. CrsMgr accepts three types of input data: integers, single words, and texts. For different types of input requests, different injection procedures are used. If non-numeric input requests contain characters with special significance in HTML these characters would be replaced by their ASCII equivalent. The replacing HTML special character procedure helps to invalidate XSS. Special characters in HTML source code are used to construct tags, URLs, and other parts of the source code of a web page and used in creating XSS. Thus, to invalidate XSS, a feasible way is to replace these special characters with corresponding alternatives [31]. CrsMgr also checks if the input contains any special characters or SQL phrases, if it does then such input requests will be discarded. To report suspicious user behavior, it is necessary to detect if the interaction is a possible attack. If the expected input is a number, then CrsMgr will check if the input is non-numeric. If the input is a single word or text, then CrsMgr will escape HTML characters then check if there are any SQL keywords or suspicious or dirty data. If the CrsMgr detects the input as suspicious (i.e., SQL keywords, special characters or unexpected data) then it will log the user's IP address and basic user information if the user is logged in and records the date and time and the trail of the validation process, and later disconnects the user.

CrsMgr using "HttpOnly" flag will prevent a cookie from being accessed using JavaScript. These are accessible when being attached to HTTP requests. It helps in the prevention of XSS attacks that expose the user's information. Each cookie has a uniquely generated session ID stored. The server checks the session ID present in the cookie and its database to verify the user. CrsMgr has features to identify XSS attacks and block such requests from executing. These XSS scripts could contain codes to steal user cookies. Most of the browsers allow users to delete cookies and deny specific types of cookies, thus protecting the web application from vulnerabilities.

3.4 Broken Authentication & Session Management

Session ID or Session Identifier or Session Token is a piece of data that is used in network communications (often over HTTP) to identify a session, a series of related message exchanges. A Session ID is necessary when the communication is using a stateless protocol (HTTP) [11].

Many websites need user credentials to log in to their website. When the user logs in with their user name and password, the website server will generate a Session ID and stores it in its database. A Session ID is a unique key generated by the webserver that helps it to identify the user. When a user is browsing through the website, the Session ID will be sent back and forth between the web server and user machine. If Session ID is not secured then the hacker can easily access the user's private information. If the website exposes the session ID in the URL then the attacker can easily access this Session ID and gain access. Hackers can also access the data if the Session ID is not encrypted or in the readable format during the communication with a web server when the user is accessing the website from a public network. If the webserver generates constant Session IDs then such a website is more vulnerable.

To prevent broken authentication and session management following rules must be observed [14].

- Implementation of multi-factor authentication to avoid brute force attack
. - The website does not allow users to use a weak password. Add restriction on length and increase the complexity of passwords by making them alphanumeric.
- Usage of strong credential recovery
- Limit the login attempts and alert the user when any such attacks happen.
- Use a session manager on the server-side to generate a random Session ID with high entropy after login. This Session ID should not be passed in URL, securely stored, and invalidated after logout, idle or absolute time-outs.
- Implement certificate authentication (SSL authentication) for a more secure channel.

The CrsMgr report has information on broken authentication and session management [3]. The web application generates a unique key to every user's session. The generation function of a key includes the user's identity that is the username that is a unique string generated by the application and Session ID based on PHP session. The SHA-256 hashing algorithm hashes the generated key.

4 INSECURE DIRECT OBJECT REFERENCES

When a website URL has reference to an internal implementation object, such as a file, directory, or database key then the occurrence it is termed as Direct Object Reference [15]. If there are no checks on such references, attackers can access unauthorized data. If there is no proper protection then the attacker can manipulate or update the data generated previously at the server-side and access the forbidden information or modify data.

To prevent an insecure direct object reference attack, it is recommended to hash the reference value in the URL. Make sure to use a

difficult hash function so that hackers cannot guess the hash algorithm. It is recommended not to use any reference values in URL and try to avoid developing such design and storing session details. Using a flow control system that controls all the data generated at the server-side, ensures the confidentiality of the data. CrsMgr uses the hash algorithm to store sensitive information[3]. The web application prevents malicious users from manipulating any data. The Session IDs of the web application are hashed and encrypted.

5 SECURITY MISCONFIGURATION

Implementation of poor or no security checks or having errors is called security misconfiguration. Attackers will often attempt to exploit non-patched flaws or access default accounts, unused pages, unprotected files, and directories, etc., to gain unauthorized access or knowledge of the system. Security misconfiguration can happen at any level of an application stack, including the network services, platform, web server, application server, database, frameworks, custom code, and pre-installed virtual machines, containers, or storage[16]. Misconfiguration attacks can also happen if any unused ports are open. This will create routes for remote attacks. Such security misconfigurations create a path for accessing unauthorized information, in turn, this will create a huge impact on business and system quality.

Currently the web and its applications are rapidly evolving and with increasing complexity. To prevent the web application from security misconfiguration following process needs to be considered [16].

- Shut-down all the unnecessary features (ports, pages, default/unused accounts, non-existing pointing applications, and remove incorrect folder permission).
- Use scanners to verify the configurations.
- Send security headers or directives.
- Review and update the configurations appropriately after necessary updates or patches installed.

6 CROSS-SITE REQUEST FORGERY (CSRF)

Cross-site request forgery is an attack that tricks an end-user into running malicious requests on a web application, and the website will not be able to identify whether the input is legitimate. CSRF attacks exploit the trust that a site has for a particular user [21]. CSRF attacks specifically target state-changing requests, not theft of data, since the attacker has no way to see the response to the forged request. If the user clicks a malicious link then it can force the user to perform state-changing requests. Attackers can also have CRSF commands in the form of image tags, hidden forms, and JavaScripts XMLHttpRequests; and these commands get executed without the user's knowledge. CSRF exploits the trust that a site has in a user's browser. CSRF is also known as XSRF, Sea Surf, Session Riding, Cross-Site Reference Forgery, one-clink attack, and Hostile Linking [17]. In 1988, Norm Hardy published a document on "confused deputy" which explains the application-level trust issue [2].

A CSRF attack uses the cookies of the authenticated user that are stored in the logs of the website to perform any state-changing operation on the website. These cookies can contain session details, and the website will consider it as an authenticated request. Cross-Site Request Forgery is a type-of confused deputy attack, as it forges the web request sent by the hacker. Many popular websites like had vulnerabilities to CSRF [5].

CSRF forges the request and can change the password or transfer money from a bank account. Therefore it is crucial to prevent web applications from such attacks. The following prevention measures need to be considered [17].

- Generate token, secret, and unique value for each request using a complex hash algorithm. Attackers will not be able to guess the correct token to forge the web request. Use these tokens to validate the request.
- Usage of cookie header token.
- An additional "SameSite" attribute can be included when the server sets a cookie, instructing the browser on whether to attach the cookie to cross-site requests.
- Validate the HTTP headers

CrsMgr generates a unique session for each transaction of the user[3]. The generation of unique and complex Session ID uses the SHA-256 hashing algorithm. Along with Session ID, even critical information such as user-name is encrypted and hashed to prevent the attack. This web application also detects hidden scripts sent along with input and blocks or does not execute such malicious requests. It also logs such malicious input requests of the user. CrsMgr uses "HttpOnly" cookies to protect from hacking the sensitive data such as Session ID.

7 EXPOSING SENSITIVE DATA (ENCRYPTION OR HASHING DATA)

Sensitive data exposure occurs when a web application exposes private data in a readable format. Whereas in a data breach, an unauthorized user steals/accesses the information from the storage. Sensitive data such as username, password, credit card details, bank account numbers, social insurance number, address, phone numbers, session tokens, and date of birth stored in the database without proper encryption or hashing will create a huge security issue. The risk of data exposure is more when the website does not use HTTPs security. The data exposure attacks can happen when the data is stored in insecure data storage in plain text.

To prevent sensitive data exposure attack the following rules should be observed.
- Use HTTPs for developing the web application.
- Never store sensitive data in a plain text. Use strong cryptographic algorithms to encrypt the data. Web application needs to identify which data is sensitive. For example all passwords need to be in encrypted format.
- Avoid storing sensitive data unnecessarily.
- Disable caching of sensitive data in plain text.
- Encrypt all data in transit with security protocols such as TLS with perfect forward secrecy (PFS) ciphers, cipher prioritization by the server, and security parameters. Enforce encryption using directives like HTTP Strict Transport Security (HSTS) [18]

The report on CrsMgr has information on the encryption of session ID parameters in a cookie file to secure the web application. The session ID is a concatenation of user-name and a unique string generated by CrsMgr, that is PHP session ID [3]. The concatenated string is hashed using the SHA-256 algorithm and later this hashed value is encrypted and stored in a cookie file. The usage of encrypted data protects from an injection. To secure the parameters' integrity and increasing the difficulty of hacking CrsMgr, parameter encryption was introduced in CrsMgr. The encryption algorithm used here is AES 256, and the length of the key used for encryption is of length 256. This algorithm generates 2^{256} key possibilities, which makes decryption almost impossible. However, the only critical parameter is encrypting to avoid performance bottleneck. CrsMgr web application uses HTTPs secure connection. In HTTPS, the communication protocol encrypts using Transport Layer Security (TLS) or, formerly, Secure Sockets Layer (SSL). HTTPS encrypts the data transferred between the client and server (bidirectional flow encryption), protecting the communication from eavesdropping. The encryption of the parameter is in process.

8 DISTRIBUTED DENIAL-OF-SERVICE

Denial-of-service (DoS) is an attacking technique where attackers flood several superfluous requests in an attempt to overload systems and prevent some legitimate requests from being fulfilled. In a distributed denial of service (DDoS) attack, the attacker floods the victim's incoming traffic requests that are originating from different sources, thus, making it difficult to stop the attack.

The first denial-of-service attack happened in 1996; Panix, the third oldest ISP, in the world was attacked first [7]. On March 5, 2018, an unnamed customer of the US-based service provider Arbor Networks fell victim to the largest DDoS in history, reaching a peak of about 1.7 terabits per second [10]. Even Wikipedia and Hong Kong's messaging app Telegram was subject to DDoS attacks [9].

There are two forms of denial-of-service attacks: those that crash the service and the other that floods the service. A distributed denial-of-service attack is most dangerous. To secure a web application from such application attacks the following rules should be observed [19].

- Prevent a single point of failure and limit input request size.
- Avoid high CPU consuming operations. Handle overflow, underflow, and exceptions.
- Avoid operations that must wait for the completion of enormous tasks to proceed. Keep queues small.
- Limit session bound information and server-side session time based on inactivity and a final time-out

There are no high CPU consuming operations in CrsMgr. There is no process in CrsMgr which waits for the huge task to complete. Session time-outs are well-developed in the CrsMgr web application. If CrsMgr identifies any input request as malicious, then it will log out the user session if logged in and logs the user's information in its database. If the page is idle for a specified amount of time, then the user's session will be logged off.

9 INSUFFICIENT LOGGING & MONITORING

Insufficient logging and monitoring are like bedrock, and hackers take advantage of this to achieve their goals without being detected. Failure logins, warnings, and errors generated need logging. Based on this logging, a web application can take actions: for example, if there are three or more login failures, the system can lock the user account temporarily and alert the authenticated user. Effective monitoring and alerting suspicious activities help to prevent insufficient logging and monitoring attacks. Most successful attacks start with vulnerability probing. Allowing such probes to continue can raise the likelihood of successful exploit to nearly 100% [20]. In 2016, identifying a breach took an average of 191 days – plenty of time for damage to be inflicted [20].

The following process can be used to prevent web applications from the exploitation of insufficient logging and monitoring [20].

- Ensure all login, access control failures, and server-side input validation failures can be logged with sufficient user context to identify suspicious or malicious accounts, and held for adequate time to allow delayed forensic analysis.
- Ensure that logs generated are in a format that can be easily consumed by a centralized log management solution.
- Ensure high-value transactions have an audit trail with integrity controls to prevent tampering or deletion, such as append-only database tables or similar.
- Establish effective monitoring and alerting such that suspicious activities are detected and responded to in a timely fashion.
- Establish or adopt an incident response and recovery plan

The report on CrsMgr has information on the logging and monitoring of a CrsMgr web application. It reports suspicious user behaviour such as malicious users using suspect inputs while accessing the web application. In this case, the system would record the input request and the user's IP address and other primary information,. The log also includes the data, time, and trail of the validation process.

10 CONCLUSION

The original aim of the internet was to provide decentralization and local autonomy while providing communication and interconnections. Some of the protocols developed, while providing flexibility, had inherent security issues. The web's transfer protocol and the initial implementation of it was concise. However, its flexibility has, also, given rise to a number of issues. DoS and DDoS should be something that each and every application should not have to contend with and should be resolved at a generic level. In view of this and the issues raised in this paper, cyber security is becoming more and more important as we become more connected and are dependent on huge tech corporations for basic services and connectivity. It is hoped that the control of these services is either completely decentralized and/or some public oversight is added to the mix

One hops that in the future until the web server has adequate DoS, DDoS detection and prevention features are built in. Web application such as CrsMgr, would need to use AI techniques to detect multiple requests of the same type either from a same or

different user (IP address) and block such requests being executed or even block the user/IP address. It is hoped that machine learning concepts are applied to make both the web and the internet address the issues raised in this paper. Usage of NLP techniques to filter keywords and predict any malicious commands passed with an input request. Web service software would evolve to detecting or blocking the execution of hidden hyperlinks to phishing websites. A mobile operating system, perhaps based on open Linux, would free users from the current duo-poly and the handset would have sufficient memory and back up mechanism to secure the owner data under the user's control. As we become more dependent on the web and mobile devices, we need to make them more secure and not hacker's heaven.

REFERENCES

[1] All about Cookies, https://www.allaboutcookies.org/cookies/
[2] Auger, Robert: The Cross-site Request Forgery (CSRF/XSRF) FAQ, https://www.cgisecurity.com/csrf-faq.html
[3] Bush, Vannevar: As we may think, The Atlantic, July 1945, https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/
[4] Capano, F. Edit this cookie, http://www.editthiscookie.com/
[5] Cross-site request forgery (CSRF), Wikipedia, https://en.wikipedia.org/wiki/Cross-site_request_forgery
[6] Cross-site Scripting, Wikipedia, https://en.wikipedia.org/wiki/Cross-site _scripting
[7] Distributed Denial of Service Attacks - The Internet Protocol Journal - Volume 7, Number 4. Cisco. https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/back-issues/table-contents-30/community.cisco.com/t5/security/ctp/4561-security
[8] Bibliography of Doug Engelbart, Doug Engelbart Institute, https://www. dougengelbart.org/content/view/163/124/
[9] Denial-of-Service (DoS) attack, Wikipedia, https://en.wikipedia.org/wiki/Denial-of-service_attack
[10] Goodin, Dan "US service provider survives the biggest recorded DDoS in history". Ars Technica. https://arstechnica.com/information-technology/2018/03/us-service-provider-survives-the-biggest-recorded-ddos-in-history/
[11] Hopgood, Bob: History of the Web, Oxford Brookes University 2001, https://www.w3.org/2012/08/history-of-the-web/origins.htm#c7
[12] Internet Engineering Task Force (IETF), HTTP State Management Mechanism, April 2011, https://tools.ietf.org/html/rfc6265
[13] OWASP, Cross Site Scripting (XSS), https://owasp.org/www-community/attacks/xss/
[14] OWASP Top Ten 2017, Broken Authentication, https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication
[15] OWASP Top Ten 2017, Insecure Direct Object Reference Prevention Cheat Sheet, https://cheatsheetseries.owasp.org/cheatsheets/Insecure_Direct_Object _Reference_Prevention_Cheat_Sheet.html
[16] OWASP Top Ten 2017, Security Misconfiguration, https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A6-Security_Misconfiguration
[17] OWASP Cross Site Request Forgery, https://owasp.org/www-community/attacks/csrf
[18] OWASP Top Ten 2017, Sensitive Data Exposure, https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure
[19] OWASP Denial of Service Cheat Sheet Article, https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html
[20] OWASP Top Ten 2017, Insufficient Logging and Monitoring, https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A10-Insufficient_Logging%252526Monitoring.html
[21] Shiflett, Chris; Cross-Site Request Forgeries, PHP Architect, Dec 2004, http://shiflett.org/articles/cross-site-request-forgeries
[22] Session ID, Wikipedia, https://en.wikipedia.org/wiki/Session_ID
[23] Using HTTP Cookies, MDN Contributors, https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies#SameSite_cookies
[24] SQL Injection, Wikipedia, https://en.wikipedia.org/wiki/SQL_injection
[25] HTTPOnly Cookie, https://owasp.org/www-community/HttpOnly
[26] How to prevent SQL injection attacks, https://www.ptsecurity.com/ww-en/analytics/knowledge-base/how-to-prevent-sql-injection-attacks/#4
[27] SQL Injection, https://portswigger.net/web-security/sql-injection
[28] As We May Think, Wikipedia, https://en.wikipedia.org/wiki/As_We_May_Think
[29] NLS or oN-Line System (computer system), Wikipedia, https://en.wikipedia.org/wiki/NLS_(computer_system)
[30] HTTP cookie, Wikipedia, https://en.wikipedia.org/wiki/HTTP_cookie
[31] W3C, 5 HTML Document Representation, https://www.w3.org/TR/REC-html40-971218/charset.html#h-5.3.2.
[32] A little history of the World Wide Web, https://www.w3.org/History.html
[33] Zhu, Jianhui: Secure CrsMgr: a course manager system, Master's thesis, Concordia University, 2016.

Spatio-Temporal Event Discovery in the Big Social Data Era

Imad Afyouni
Computer Science Department,
University of Sharjah
Sharjah, UAE
iafyouni@sharjah.ac.ae

Aamir S. Khan
Faculty of Computer Science,
Dalhousie University
Halifax, Canada
aamirkhan.wc@gmail.com

Zaher Al Aghbari
Computer Science Department,
University of Sharjah
Sharjah, UAE
zaher@sharjah.ac.ae

ABSTRACT

Social networks have been transforming the way people express opinions, post and react to events, and share ideas. Over the last decade, several studies on event detection from social media have been proposed, with the aim of extracting specific types of events, such as, social gatherings, natural disasters, and emergency situations, among others. However, these works do not consider the continuous processing of events over the social data streams, and therefore, cannot determine the spatial and temporal evolution of such events. This paper introduces a big data platform for event discovery, while tracking their evolution over space and time. We propose a scalable and efficient architecture that can manage and mine a huge data flow of unstructured streams, in order to detect geo-social events. The extracted clusters of events are indexed by a spatio-temporal index structure. We conduct experiments over twitter datasets to measure the effectiveness and efficiency of our system with respect to the existing major event detection techniques. An initial demonstration of our platform highlights its major advantage for detecting and tracking events spatially and temporally, thus allowing for great opportunities from application perspectives.

CCS CONCEPTS

• **Information systems → Data mining**; **Spatial-temporal systems**; **Data management systems**; • **Computing methodologies → Natural language processing**;

KEYWORDS

Social Big Data, Event Detection, Spatio-Temporal Scope, Data Stream Management

ACM Reference Format:
Imad Afyouni, Aamir S. Khan, and Zaher Al Aghbari. 2020. Spatio-Temporal Event Discovery in the Big Social Data Era. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410566.3410568

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410568

1 INTRODUCTION

Over the last two decades, social media has emerged as a great support for understanding users' and community behavior. Mining social data sources, such as twitter and Instagram, allows for fruitful insights to be developed about online interacting users. People discuss almost all topics of interest on social networks, and use them to post their daily life events, share some feedback, and/or add check-ins of their visited places, among others. Analyzing user-generated content can provide insights about surrounding live events of interest (EoI) or any unusual happenings, given that a lot of relevant spatio-temporal information is embedded in social media streams.

In the context of social media, an 'event' can be defined as the occurrence of a real-world unusual happening within a specified space and time. For example, a spike in the number of tweets talking about a new festival inauguration in Paris, coupled with related hashtags around nearby places, would definitely create a new local attraction with a higher significance within that spatio-temporal extent. Social events of interest usually include gatherings, concerts, incidents, job announcements, or natural disasters, among others (see Figure 1). Detecting such events in real-time can leverage new ways for exploring current cities. The dynamic updates of such events by the live communities in social media lay the ground for developing plenty of intelligent location-based services that can play a critical role in tourism, navigation, city exploration, education, and emergency management, among others.

Existing studies on event detection aim at detecting specific or generic types of events, but generally focus on extracting the main topic and subtopics of such events. However, extracting other features that describe the evolution and spanning of such events over space and time need further investigation. For instance, small-scale events (e.g., accident or traffic jam) may be alive for a few hours, whereas large-scale events (e.g., storm or election) may span for several days or weeks, and over a larger geographical extent. Furthermore, existing event detection systems do not fully support big data stream processing, which is mandatory to achieve a scalable and worldwide event extraction and visualization.

In this paper, we introduce a new perspective for event discovery in which the spatio-temporal evolution of dynamic social events is detected and tracked events. We present a big data mining platform for the discovery of geo-social events. These detected events are tagged with spatial and temporal components. A detected new event is either associated with an existing cluster or becomes a seed of new cluster. The system is featured with a scalable and efficient architecture that can manage and mine huge data flow of unstructured data streams such as Twitter. The clustered events are indexed by a spatio-temporal index structure. Therefore, our

system leverages the spatial and temporal tracking of the detected social events. This allows for deep understanding and enhanced emergency responses in diverse uses cases such as innovative and distinctive tourism services, but also in analyzing and monitoring natural disasters and spreading of epidemic diseases. Indeed, this platform leverages the monitoring and prediction of such events based on discovered patterns and user behavior.

The rest of this paper is organized as follows. Section 2 briefly discusses the related work, while Section 3 presents the system overview and the main architecture components. Section 4 introduces the details behind the spatio-temporal event detection technique, while Section 5 highlight some implementation details. Section 6 presents the evaluation and discussion on results. Finally, concluding remarks are highlighted, showing the potential of this research.

2 RELATED WORK

Event of Interest (EOI) Mining

Many real-world applications such as for traffic updates, surveillance, natural disasters, and epidemic diseases, continuously collect data to detect and track such unusual happenings. Early event and anomaly detection from social media can be helpful to society, users, and authorities to take proper action on time [4, 5]. Social data streams can be observed as the representation of the real-world happenings at a certain location and time. These happenings can be classified based on the Thematic, Temporal, Spatial, and other learning features (e.g., user profiles and social links). We propose to asses social event extraction techniques based on the following dimensions: i) type of the event, whether specific (e.g., earthquakes) or generic; ii) classification model, whether supervised, unsupervised learning, or none; and iii) learning features: spatial features, temporal features, textual features, and semantic or contextual features of the user-generated content.

Discovering and disseminating events over online social networks have been heavily studied in the last decade. For example, several studies have used twitter data to predict different types of events such as earthquakes [15], soccer games [11], road traffic [1] and diseases like influenza [2]. Research of event detection originates from the area of Topic Detection and Tracking (TDT) [3]. Exploring other data representation techniques and features was also performed by McMinn and Jose (2011) [10], Unankard et al. (2015) [17] (LSED), Kaleel and Abhari (2015) [7]. Online clustering and NaÃŕve Bayes classifiers applied by Jagan Sankaranarayanan [16] and proposed a TwitterStand framework. Hasan et al. [7] have proposed approach a similarity threshold-based clustering and implement a framework named TwitterNews+. Type-aware Bias Neural Network with Attention Mechanisms (TBNNAM) [9] to reduce manual effort in detecting events by using without triggers.

The authors in [20] have used geotagged data from Flickr to analyze and identify the occurred events by extracting their type, location and time. Geotagged data has also been used to find clusters of events in large spatial databases [6, 21]. In a companion work [12–14], we have also developed techniques to enrich digital maps with live constraints and multi-scale events such as accidents, road closed, and traffic flow speed from social media data. Although their work considers spatial extent detection in a hierarchical manner, but the approach developed is not incremental, does not consider the temporal evolution of events, and uses a supervised learning method, i.e., bag of words, to detect events of interest. In this paper, with respect to the related work, we take advantage of the state-of-the-art big data stream management technologies, to build a fully fledged system that incorporates a spatio-temporal event social detection technique.

Performance and Scalability Perspectives

Data stream processing leverages a continuous manipulation and execution of unbounded streaming data. Extracting effective information from large datasets is an essential characteristic of big data-oriented applications. Social big data applications can be enlarged to a wide number of domains, and the experimental evaluation needs to focus on efficiency and real-time performance, in addition to the correctness and accuracy of results. Many proprietary data stream management systems (DSMS) are available in the market such as Google Cloud Dataflow, IBM Streams, Amazon Kinesis, among others. On the other hand, open-source frameworks for data stream processing include Spark Streaming, Apache Storm, Elastic Search, Apache Kafka, and Flink. In general, data streaming processing means a continuously process to manipulate unbounded streaming data coming from real-time sources. Beyond data stream processing, an efficient management and retrieval of detected events is required. Therefore, a spatio-temporal indexing structure is needed, which allows fast retrieval and updates on existing clusters, in order to accommodate new streaming data. Many recent attempts were introduced to integrate spatial and spatio-temporal indexing in non-relational distributed databases [18, 19]. Consequently, an approach that integrates an efficient data stream processing, along with a robust multidimensional indexing scheme for managing evolving events of interest is going to be developed towards building a highly scalable social event detection system.

3 SYSTEM OVERVIEW

This section presents a novel map-based platform that collects social data streams from social networks, processes data to find events of interest (EoI) and visualizes detected events on maps. There are multiple components in the proposed big data pipeline. Figure 1 illustrates an overview of our event extraction technique with the salient components. The main components of our system architecture are highlighted as follows.

- Data Acquisition: It involves gathering data from unstructured social media data. Digesting data streams is performed by running crawlers that collect bulks of streams. Although this paper focuses on only one source of data (i.e. Twitter), an integration of other sources is possible by designing new crawlers and generating new streaming packets as input to our framework.

- Data Preprocessing & Ingestion: The preprocessing steps include filtering and packaging streams into small bulks of data as continuous streaming windows, thus allowing to process data within a short temporal slice, while taking results from historical windows into account. The Twitter API filters like language, location, etc. is being used to ignore the unwanted tweets. The data is packaged in a specific JSON format with the required fields. The packaged

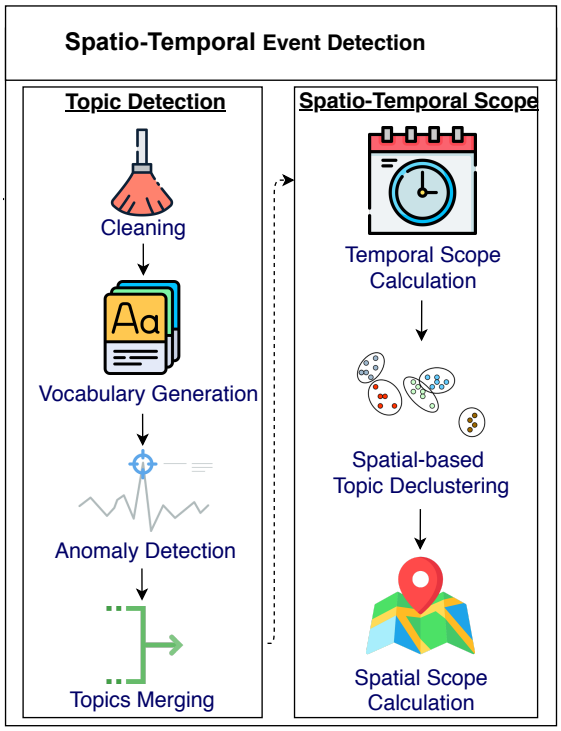


Figure 1: Event Extraction Process

tweet is ingested to an Apache Kafka topic, before generating the data stream window for the current time slice.

- In Event Detection, we classify and extract events within different categories, such as, social events (e.g., concert, match, graduation party), road accidents, incidents, accident prone areas, non-safe area, and other breaking news. Each detected event has an associated spatial and temporal component, which is also calculated in this component. This component works continuously to process data from incoming streams in order to extract new clusters or update existing clusters with new packets.

- Ingestion and Indexing of Spatio-Temporal Events: The events detected in the previous step are stored in a spatio-temporal indexed database. This component supports the efficient retrieval of events based on the main querying attributes, spatial and temporal information.

- Visualization: The Visualization component provides a new dimension to existing maps by illustrating extracted knowledge from live streams in the form of live events with different spatial and temporal scopes.

The next sections present details on the process flow and algorithms towards building the big data platform for social event detection.

3.1 Data Preprocessing & Ingestion

The data streams are collected from Twitter. Other sources such as Instagram and flicker will be considered in our future research, so that multi-modal data streams can be integrated with twitter text streams. The Data Preprocessing and Ingestion phase involves three major steps: Filtering, Packaging and Ingestion.

- Filtering: Filtering plays a key role in getting the right data. In our case, we ignore the tweets that do not match our criteria. The Twitter Streaming API allows us to filter tweets using language, location, etc. parameters. We have ignored the tweets that are not English using the language parameter. Since an event has a spatial and temporal attributes associated with it, we have only considered geo-tagged tweets.

- Packaging: The packaging step involves extracting the required fields for event detection. The fields chosen are: $created_{at}$: This is the date-time instance of the tweet creation time

id_{str}: This is the unique ID of the tweet as string

$text$: This is the tweeted text.

$coordinates$: These is point coordinate. It includes Latitude and Longitude.

- Ingesting: We used Apache Kafka to build a fault-tolerant big data pipeline. The Kafka producer will read streams from twitter and publish it to a Kafka topic. The data from Kafka is consumed in real-time. We have used Python to write a Kafka Consumer script.

The twitter payload is packaged with the required fields into a Kafka payload. Kafka payload is published to a Kafka topic. Each Kafka payload has the following JSON format:

date: '2019-10-05 16:42:55', *'latitude'*: 37.32034997, *'longitude'*: -122.00979481, *'text'*: "I'm at Cupertino High School in Cupertino, CA. https://t.co/vSgoVzuzgC", '$tweet'_{id}$: '118052377565952'

This published payload is later consumed Kafka to perform event detection. Kafka makes it possible to perform all of the above steps in near real-time. To add new source in our system, we just need to add small piece of code with out impacting other components of the system as kafka allows multiple topics. Each topic in kafka can be consumed separately and concurrently.

- The detected events are depicted by the following fields:

$event - id$: indexed unique identifier representing an event

$topics$: keywords that represent an event

$tweet - ids$: identifiers of tweets

$tweet - texts$: texts of tweets

$importance - score$: magnitude of the event

$start - date$: start date and time of the temporal scope of event.

$end-date$: temporally indexed start date and time of the temporal scope of event.

$Duration$: the duration or lifetime of an event.

$geom$ (longitude, latitude): spatially indexed point coordinates

The final event clusters are based on a combination of topics, temporal evolution and spatial clustering. Spatial clustering has a significant impact in our event detection process. The clustering parameters include: i) the minimum cluster size, which is the smallest size of groups to be considered for a cluster; and ii) the cluster radius, which helps in merging clusters within a specified distance.

3.2 Event Detection

Event Detection classifies and extracts events within different categories, such as, social events (e.g., concert, match, graduation party), road accidents, incidents, accident prone areas, non-safe area, and

other breaking news. The event detection component works continuously to process data from incoming streams in order to extract new clusters or update existing clusters with new packets. Event detection consists of two major steps:

3.2.1 Topic Detection. Topic detection is the fundamental step of event detection. The burstiness of a word allows us to classify if a word is a topic or not. Unspecified topic detection method involves extraction of spatio-temporal spikes and unusual happenings based on the top frequent words. Here, each word that is considered for topic detection is part of the tweet text.

Before the topic detection process starts, tweets are read from the Kafka topic where we packaged and stored all the tweets. Tweets are read periodically using a certain time interval that the user provides. Topic detection is performed on each batch in near real time, thus maintaining up to date files on word frequencies and tweet corpus.

The topic detection process uses various Natural Language Processing (NLP) techniques to process the tweets and get the topics. The detection process works as follows:

- The *cleaning* phase removes all the unnecessary characters or words from the tweet text. Tweet text is split into words using the concept of tokeninzation and irrelevant words are filtered out using stop words. A stop word is a commonly used word (such as 'the', 'a', 'an', 'in') that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the result of a search query. Since these words cannot be considered as topics, these stop words are used to filter out these words from the tweets.

- Vocabulary generation: This is a very important phase where it picks out the words that have the potential to become a Topic. After the cleaning step, each word in the batch goes through this phase. Words and their associated counts are stored in json objects. These words are also given an id and are store in another json object, VOC. VOC is the vocabulary.

- Anomaly detection: Before formulating the anomaly measure, they define the expected number of mention creation associated to a word t for each time-slice i. They assumed that the number of tweets that contain the word t and atleast one mention in the ith time-slice, follows a generative probabilistic model. Thus they computed the probability P(mention in the ith time-slice) of observing mention in the ith time-slice. For a large enough corpus, it seems reasonable to model this kind of probability with a binomial distribution. Moreover, since N is large they further assume that P(mention in the ith time-slice) can be approximated by a normal distribution. The Anomaly is calculated for each word in each of tsb time-slices where tsb is inputted by the user. The sum of all the anomaly scores across tsb time-slices gives a Magnitude of impact (MAG) of the topic. The topics that have a 0 MAG score are ignored and are not considered as a topic.

- Topic Merging: The topics detected so far are the basic or first level topics. Some of these topics may point to the same event. The work presented in [6], was focused on a mention-anomaly-based approach for topic detection, referred to as MABED. This has a way of merging these similar topics into a single list of topics. It involves the following steps:

• Identification of the candidate words
• Selection of the most relevant words
• Generating the list of right topics
• Detecting duplicated topics
• Merging duplicated topics

Initially p candidate words are extracted from the tweet text. These candidate words are a topic's co-occurring words. These candidate words are further filtered out based on the similarity threshold parameter Îÿ (theta). Finally the duplicated topics are merged. The topic detection used in this paper was inspired by the work presented in [6]. However, MABED approach does not take the spatial dimension into account, and does not consider the real-time processing of incoming streams. Thus, the topic detected in MABED was a generic topic where anyone from around the world may be talking about. However, our approach extract real spatio-temporal events that occurred within a geographical extent, and have a temporal evolution attached to them.

3.2.2 Spatio-Temporal Scope. Spatial and Temporal scope extraction: For a topic to be an event, it must have an associated spatial and temporal information. Spatial information is the location of the event, whereas, the temporal information includes a start and an end date, and the duration of that event. We consider Topic-related tweets as an event. These tweet events have the same temporal information as of the Topic. However, the Spatial information may be different.

Temporal Scope Calculation: Using the anomaly scores calculated for each topic, the temporal scope is estimated. The temporal scope is the Time-To-Live (TTL) factor of the event. Every topic and event will have a start timestamp (date and time) and an end timestamp. Since, the number of anomaly scores calculated for each topic is equal to the tsb time-slices, the temporal scope is estimated between these two time-slices.

Spatial Scope Calculation: The spatial scope used to extract geographic content's source latitude, longitude coordinates for identifying places, organization, city name, street names, and landmarks, etc. The spatial aspect of location-based social networking services strengthens the connection between social networking services and real-world social networks. Spatial Information extraction step comes after the topics are detected and merged. Since all the tweets are geo-tagged, these point coordinates help us in estimating the coordinates. Each topic has related tweets and these tweets may have different coordinates. For eg: Topic 'job' may have five different tweets and all of these tweets talk about job. However, they may talk about different jobs at different locations. Here, each different job with different location can be considered as a separate event. Secondly, we apply a spatial clustering algorithm on the coordinates of all the tweets to cluster all the tweets in close proximity. We have used HDBSCAN clustering algorithm as HDBSCAN supports haversine distance (i.e. longitude/latitude distances) which will properly compute distances between geo-locations. This process is called de-clustering because it is separating the tweets that belong to one topic using the spatial clustering.

$events = \{$
$attributes = [$
$name = `tweet_ids', type = `String[]', index = true$
$name = `tweet_texts', type = `String[]', index = false$
$name = `dtg', type = `Timestamp', index = false$

$name = 'TTL', type = 'Integer', index = false$
$name = 'geom', type = 'Point', index = true, srid = 4326$
] }

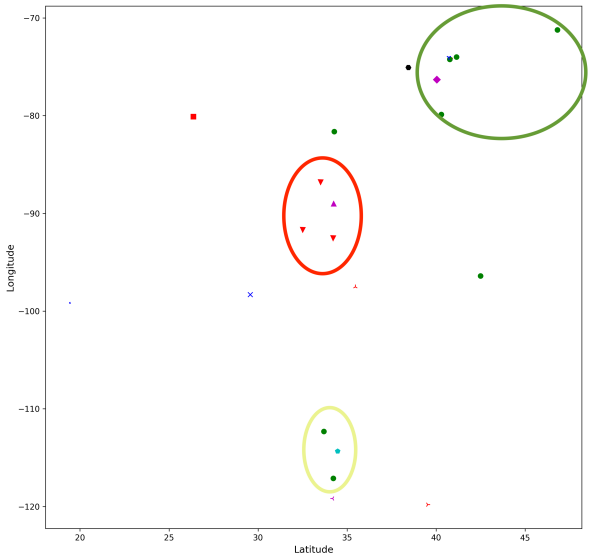


Figure 2: Spatial Clusters of Detected Events

From the above process, we get spatial clusters and topic clusters (see Figure 2). Each of these spatial clusters have an id called spatial cluster id. An event is formed by grouping the tweets by *topic* and *spatial cluster id*. Each of these groups is assigned an id called the event id. These detected events are published to another topic in Kafka in a continual manner for the pipeline to work smoothly.

3.3 Big Data Stream Management

Once the events are detected, we need to find out if an event is a new event. If the detected event already exists in the database, we may need to update the spatio-temporal property of the existing event in the database. Otherwise, we create a new event. New events are published to a Kafka topic and existing ones are updated in the database.

To build a fast querying pipeline, a spatio-temporal indexing approach is a must. Geomesa is an open source suite of tools that enables large-scale geospatial querying and analytics on distributed computing systems. It provides spatio-temporal indexing on top of the Hbase, Accumulo, Google Bigtable and Cassandra databases for massive storage of point, line and polygon data. We have used Geomesa to build a strong big data pipeline. We have used Accumulo with Geomesa as Geomesa has the most stable support for Accumulo database.

The Ingestion of Spatio-Temporal Indexed Events phase involves two major steps: Events Cluster Integration and Events Ingestion to Spatio-Temporal Indexed Database. Once we get the active list of events then we can update the event properties in the Geomesa database with the new detected properties. This step corroborates that the detected event is actually an event and that's why this step plays a very important role. After the Events cluster integration, we spatio-temporally index the newly detected events to Geomesa-Accumulo database.

3.4 Event Visualization

After the events are ingested with spatio-temporal indexing in Geomesa-Accumulo database, we can visualize these events using Geoserver. Our big data pipeline includes Geoserver that helps us in querying and visualizing the events. Geomesa has a Geoserver plugin that makes it very simple to use Geoserver with Geomesa data.

4 IMPLEMENTATION DETAILS

The proposed system was setup on Ubuntu 18.04 operating system in a standalone mode, with an Intel Core i7-6700HQ processor and 16GB DDR4-2133 RAM. The proposed system is fully based on open source tools and libraries (e.g., Apache Kafak and Nifi, Hadoop, Geomesa-Accumulo). We use Twitter's streaming API to collect data in real-time. The streaming api is quite different from the REST API, because the REST API is used to pull data from twitter but the streaming API pushes messages to a persistent session. This allows the streaming api to download more data in real time than could be done using the REST API. The following tools are used for the ingestion pipeline:

Apache Kafka and Apache NiFi:

I. Since we already use a topic to store streaming tweets in Apache Kafka, we added another topic, event, to store newly detected events.

II. NiFi was built to automate the dataflow between systems. The term 'dataflow' here means, the automated and managed flow of information between systems.

III. In our proposed system, the data flows from event Kafka topic to Geomesa with Accumulo backend. NiFi controls this dataflow.

IV. The Geomesa-Accumulo database schema is defined before any data is transferred from Kafka to Geomesa through NiFi.

V. Accumulo: A Non-relational Distributed Database. It was inspired by Google's BigTable implementation.

VI. Geomesa: For an efficient spatio-temporal querying pipeline, we used Geomesa. GeoMesa is an Apache licensed open source suite of tools that enables large-scale geospatial analytics on cloud and distributed computing systems, letting you manage and analyze the huge spatio-temporal datasets that IoT, social media, tracking, and mobile phone applications seek to take advantage of today.

5 EVALUATION & DISCUSSION

In this section, we present the evaluate of our proposed system and the disparate experiments conducted with different parameters to evaluate our proposed system. For evaluation purposes, we collected 3,122,304 tweets during three days in December, 2019. The datasets we have collected are of much bigger size, but to facilitate verification of results, only a smaller portion of data was considered to calculate the accuracy of our approach. We only considered

Approach	Precision	Recall	F1	Clustering Accuracy
Our System	0.88	0.85	0.86	0.95
SnowChallenge	0.98	0.34	0.50	NA
MABED	0.66	0.77	0.71	NA

Table 1: Precision, Recall, F1-Score, and Clustering Accuracy

geo-tagged tweets from USA for these experiments. Additionally, we only considered english language tweets for our experiments.

For the accuracy measures, we randomly took some event clusters and manually annotated the ground truth data. For Efficiency, we allowed our big data system collect and process tweets in real-time. This real-time tweets collection and processing simulated the real-world application of our big data system.

We compared our approach with two other approaches that have some similarities, by testing their codes and reproducing the results on the same datasets: the snow challenge winner [8], which detects events with spatial coordinates using aggressive filtering and hierarchical clustering, and MABED [6] where topic detection has been the major focus based on anomaly detection. We computed the precision, clustering accuracy and F1 score for each approach.

The results show that the aggressive filtering approach presents a very high precision, but fails to score well in recall and F1 measures due to the huge loss of false negatives, which is caused by aggressive filtering. MABED, on the other hand, scores average on both precision and recall, due to the fact that the approach aims at detecting high level topics rather than spatio-temporal events. Therefore many real life events can be merged within one topic, thus reducing its accuracy. our approach scores well in the different measures, and can compete with the best approaches even though there exists no benchmark where we compare all approaches on the same basis. The clustering accuracy measure reflects the quality our extracted clusters in terms of how many tweets in the cluster are actually talking about the same event. The results show that our clustering technique works very well and the number of outliers within clusters is very low.

6 CONCLUSION

This research paper develops a big data mining platform for the discovery of spatio-temporal events from social media. The system is based on a scalable and efficient big data platform that should manage and mine a huge data flow of unstructured data streams by relying on the state-of-the-art big data and stream management tools (Spark, Kafka, Apache Nifi, geoServer, etc.). An unsupervised machine learning and NLP techniques were employed in this research for the continuous event detection, and a spatio-temporal indexing scheme was implemented for the fast retrieval of evolving events. An extensive evaluation of the developed platform will be investigated with respect to effectiveness and scalability perspectives. Several domain applications can be developed on top of this platform, such as, smart trip planning, and forecasting and tracking of natural disasters, among others.

REFERENCES

[1] B. Alkouz and Z. Al Aghbari. Snsjam: Road traffic analysis and prediction by fusing data from multiple social networks. *Information Processing & Management*, 57(1):102139, 2020.

[2] B. Alkouz, Z. Al Aghbari, and J. H. Abawajy. Tweetluenza: Predicting flu trends from twitter data. *Big Data Mining and Analytics*, 2(4):248–273, 2019.

[3] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 37–45, 1998.

[4] F. Atefeh and W. Khreich. A survey of techniques for event detection in twitter. *Comput. Intell.*, 31(1):132–164, Feb. 2015.

[5] A. Boettcher and D. Lee. Eventradar: A real-time local event detection scheme using twitter stream. In *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications*, GREENCOM '12, pages 358–367, Washington, DC, USA, 2012. IEEE Computer Society.

[6] A. Guille and C. Favre. Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach. *Social Network Analysis and Mining*, 5(1):18, 2015.

[7] M. Hasan, M. A. Orgun, and R. Schwitter. Twitternews+: A framework for real time event detection from the twitter data stream. In *8th International Conference on Social Informatics, SocInfo 2016*, pages 224–239. Springer, Springer Nature, 2016.

[8] G. Ifrim, B. Shi, and I. Brigadir. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. ACM, 2014.

[9] S. Liu, Y. Li, F. Zhang, T. Yang, and X. Zhou. Event detection without triggers. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 735–744, 2019.

[10] K. Massoudi, M. Tsagkias, M. de Rijke, and W. Weerkamp. Incorporating query expansion and quality indicators in searching microblog posts. In *Proceedings of the 33rd European conference on Advances in information retrieval*, pages 362–367. Springer-Verlag, 2011.

[11] M. Musleh. Spatio-temporal visual analysis for event-specific tweets. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1611–1612, 2014.

[12] F. U. Rehman, I. Afyouni, A. Lbath, and S. Basalamah. Understanding the spatio-temporal scope of multi-scale social events. In *Proceedings of the 1st ACM SIGSPATIAL Workshop on Analytics for Local Events and News*, pages 1–7. ACM, 2017.

[13] F. U. Rehman, I. Afyouni, A. Lbath, S. Khan, and S. Basalamah. Building socially-enabled event-enriched maps. *GeoInformatica*, 24(2):371âĂŞ–409, 2020.

[14] F. U. Rehman, I. Afyouni, A. Lbath, S. Khan, S. M. Basalamah, and M. F. Mokbel. Building multi-resolution event-enriched maps from social data. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017.*, pages 594–597. OpenProceedings.org, 2017.

[15] T. Sakaki, M. Okazaki, and Y. Matsuo. Tweet analysis for real-time event detection and earthquake reporting system development. *IEEE Transactions on Knowledge and Data Engineering*, 25(4):919–931, 2012.

[16] H. Samet, J. Sankaranarayanan, M. D. Lieberman, M. D. Adelfio, B. C. Fruin, J. M. Lotkowski, D. Panozzo, J. Sperling, and B. E. Teitler. Reading news with maps by exploiting spatial synonyms. *Communications of the ACM*, 57(10):64–77, 2014.

[17] S. Unankard, X. Li, and M. A. Sharaf. Emerging event detection in social networks with location sensitivity. *World Wide Web*, 18(5):1393–1417, 2015.

[18] M. A. Whitby, R. Fecher, and C. Bennight. Geowave: Utilizing distributed key-value stores for multidimensional data. In *International Symposium on Spatial and Temporal Databases*, pages 105–122. Springer, 2017.

[19] J. Yu, J. Wu, and M. Sarwat. Geospark: A cluster computing framework for processing large-scale spatial data. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 70. ACM, 2015.

[20] M. Zaharieva, M. Zeppelzauer, and C. Breiteneder. Automated social event detection in large photo collections. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 167–174, 2013.

[21] R. Zhua, C. Zuoa, and D. Lina. Research on event perception based on geo-tagged social media data. In *Proceedings of the ICA*, volume 2, pages 1–8, 2019.

Data Science for Healthcare Predictive Analytics

Carson K. Leung*
Dept. of Computer Science
University of Manitoba
Winnipeg, MB, Canada
kleung@cs.umanitoba.ca

Daryl L.X. Fung
Dept. of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Saad B. Mushtaq
Dept. of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Owen T. Leduchowski
Dept. of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Robert Luc Bouchard
Dept. of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Hui Jin
Dept. of Computer Science
University of Manitoba
Winnipeg, MB, Canada

Alfredo Cuzzocrea
iDEA Lab
University of Calabria
Rende, Italy

Christine Y. Zhang
Rady Faculty of Health Sciences
University of Manitoba
Winnipeg, MB, Canada

ABSTRACT

Big data are everywhere nowadays. Many businesses possess big data for their success because big data are very useful and are considered as new oil. For instance, big data are very important in predicting the trends on what will happen in the future. Many researchers have generated or gathered data to further enhance their research and to apply them to numerous real-life applications. Examples of big data include healthcare patient data. To improve the detection of illnesses and diseases, researchers have gathered healthcare patient data, examined the diagnosis on healthcare patient data (e.g., cells, blood count, antibodies count), and compared with previous data to determine if a specific illness or disease exist. Having an automatic predictive method for healthcare and disease analytics would be desirable. In this paper, we focus on healthcare mining, which aims to computationally discover knowledge from healthcare data. In particular, we present a data science framework with two predictive analytic algorithms for accurate prediction on the trends of cancer cases. The algorithms predict cancerous cells based on the information of the cell data from some data samples. Evaluation results on several real-life datasets related to the breast cancer demosntrate the effectiveness of our data science framework and predictive algorithms in healthcare data analytics.

CCS CONCEPTS

• **Information systems** → **Data mining**; *Clustering and classification*; • **Applied computing** → **Health informatics**; *Health care information systems*; • **Computing methodologies** → *Machine learning algorithms*; *Supervised learning by classification.*

KEYWORDS

Data science, healthcare data, database engineering and applications, data mining, machine learning, unsupervised and supervised learning, prediction, neural network, autoencoder, few-shot learning (FSL)

ACM Reference Format:
Carson K. Leung, Daryl L.X. Fung, Saad B. Mushtaq, Owen T. Leduchowski, Robert Luc Bouchard, Hui Jin, Alfredo Cuzzocrea, and Christine Y. Zhang. 2020. Data Science for Healthcare Predictive Analytics. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–18, 2020, Seoul/Incheon, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410598

1 INTRODUCTION

With advancements in information and communication technology (ICT), huge amounts of data have been generated and collected at a rapid rate in numerous real-life application and service domains—including database engineering and applications. Due to the nature of data and how these *big data* [27, 34] are generated or collected, these big data can be of different levels of varacity (e.g., precise, imprecise and uncertain data [20, 25]). Embedded in the big data are implicit, previously unknown, and potentially useful information and valuable knowledge. This calls for *data science*. It aims to apply database techniques, engineering and scientific methods, mathematical and statistical models, machine learning (e.g., deep learning) techniques, and/or data mining algorithms [3, 9, 16, 21, 22] to the big data for

- management of these big data;
- information extraction and knowledge discovery from the well-manged data; and
- interpretation and visualization of extracted information and discovered knowledge.

*Corresponding author: kleung@cs.umanitoba.ca (C.K. Leung)

IDEAS'20, August 12–18, 2020, Seoul, South Korea

ACM ISBN 978-1-4503-7503-0/20/08…$15.00
https://doi.org/10.1145/3410566.3410598

Examples of big data include social media data [7, 17, 26, 35], financial data [24], and healthcare data [13, 38]. Having a data science solution for analyzing healthcare data would save patient life and improve quality of our life. To elaborate, it is the saddening to learn that, according to the World Health Organization (WHO)[1], cancer is the second leading cause of death globally (with cardiovascular diseases begin the leading cause of death globally). It was responsible for an estimated 9.6 million deaths in 2018. In other words, about 1 in 6 deaths was due to cancer. There are many different forms of cancers. Some are common (e.g., lung cancer, breast cancer, colorectal cancer), and some are rare (e.g., lymphoma cancer, thyroid cancer). For instance, in 2018, there were 2.09 million cases of breast cancer, which led to 627 thousand deaths globally. Cancer is generally caused by the transformation of normal cells into tumour cells in a multi-stage process that progresses from a pre-cancerous lesion to a malignant tumour. The transformation could be triggered by:

- genetic inheritance,
- aging, and
- external agents such as:
  - physical carcinogens like ultraviolet and ionizing radiation;
  - chemical carcinogens like asbestos and tobacco smoke;
  - biological carcinogens like infections from some viruses and bacteria.

Currently, about 30%-50% of some forms of cancer could be prevented by:

- reducing the cancer burden (e.g., avoiding tobacco use, intaking healthy diet, doing sufficient amounts of physical activities regularly);
- detecting the cancer early (e.g., through early diagnosis, screening); and
- receiving proper treatments.

Moreover, having a better understanding (e.g., causes, development) of cancer would be helpful to researchers, healthcare provideres, and decision makers (e.g., policy makers). Hence, in this paper, we focus on *healthcare data analytics*. We design and develop a *data science solution for healthcare predictive analytics*.

In the health science and medical field, data are one of—if not the most important—resources for the researchers. Analysts look for trends, patterns and commonalities within and among samples (e.g., cancer strains). However, samples are expensive to obtain. Rarer strains (or insufficient numbers of samples) are more difficult for these researchers to properly utilize because they may not appear to contain sufficient (or statistically signficiant number of) data for accurate predictions. In this paper, we demonstrate that our data science solution requires only a small amount of data with certain common formulae, which combined together could successfully emulate results of the same formulae on a big dataset. Moreover, we also utilize those formulae on the rarer strains (which may not have the option of a larger dataset) with an aim to make accurate predictions.

Our *key contributions of this paper* include the design and development of our data science solution for healthcare predictive analytics. Specifically, the solution consists of a framework with two algorithms for few-shot cancer predictive analytics. Evaluation results show the our solution—which requires only a small training sample—surpasses many baseline machine learning algorithms requiring big training data in producing accurate predictions.

The remainder of this paper is organized as follows. Next section provides background and related works. In Section 3, we describe our data science solution, which consists of a framework and two algorithms for healthcare predictive analytics. Evaluation results and conclusions are shown in Sections 4 and 5, respectively.

2 BACKGROUND AND RELATED WORKS

Data mining and data analytics have been applied in numerous studies for cancer research. For instance, researchers have used open datasets on breast cancers (which capture patient clinical records like tumor radius, texture, smoothness, etc.) from UC Irvine Machine Learning Repository [15]:

- Breast cancer Wisconsin (original) data set (aka original *Wisconsin breast cancer database* (*WBC*))
- Breast cancer Wisconsin (diagnostic) data set (aka *Wisconsin diagnostic breast cancer database* (*WDBC*))
- Breast cancer Wisconsin (prognostic) data set (aka *Wisconsin prognostic breast cancer database* (*WPBC*))

Regarding related works, Rumelhart et al. [31] learned internal representations by error propagation. They built an autoencoder architecture in the reconstruction of input learning. In general, an *autoencoder* [1, 19, 39] is an artificial neural network (ANN) built for learning efficient data codings in an unsupervised fashion. To learn a data encoding—aka a representation of a set of data—(e.g., for dimensionality reduction), the autoencoder trains the network to ignore signal "noise". In addition to reduce the dimensionality, the autoencoder also reconstructs—from the reduced data encoding—a representation as close as possible to its original input.

An autoencoder can be extended to become a *deep autoencoder* [41]. Dolgikh [14] applied spontaneous concept learning with a deep autoencoder (dAEN). A deep autoencoder can compose of two symmetrical deep-belief networks, which commonly consist of 4-5 shallow layers representing the encoding half of the networks and 4-5 other layers repesenting the decoding half.

To helps machine learning models to achieve better performance, *multitask learning* (*MTL*) [4, 40] exploits commonalities and differences across multiple learning tasks to come up with a shared representation (for feature or representation learning), and solves these tasks in parallel by using the shared representation. Moreover, by using the domain information contained in the training signals of related tasks as an inductive bias, MTL improves generalization through inductive transfer. In other words, it applies the knowledge learned from a task to improve the the learning of other tasks.

There have works on predictive analytics on various domains [2, 29]. For works related to breast cancer predictions, Rani [30] used a neural network technique to achieve 92% accuracy with 300 training samples and 50 test samples. As neural networks are generally capable of learning complex and non-linear relationships including noisy or less precise information, they are well suited in biomedical engineering. To speed up the training process for classification, Rani adopted a parallel approach. In contrast, Choi

[1] https://www.who.int/news-room/fact-sheets/detail/cancer

et al. [6] proposed a hybrid Bayesian network (BN) model by combining the ANN and the Bayesian network (BN) to predict breast cancer prognosis. Their model outperformed the baseline BN model with a high degree of performance inherited from ANN while retaining the good explanation power from the BN. Experimental results show that their model achieved a prediction accuracy of 87.2% with a sensitivity of 93.3% and a specificity of 83.1%. Note that (i) *accuracy*, (ii) *sensitivity* (aka *recall*, hit rate, or true positive rate (TPR)), (iii) *specificity* (aka selectivity or true negative rate (TNR)), (iv) *precision*, and thus (v) *F1 score* can be computed as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$recall = sensitivity = \frac{TP}{TP + FN} \quad (2)$$

$$specificity = \frac{TN}{TN + FP} \quad (3)$$

$$precision = \frac{TP}{TP + FP} \quad (4)$$

$$F1score = \frac{2TP}{2TP + FP + FN} \quad (5)$$

where

- TP are true positives,
- TN are true negatives,
- FP are false positives, and
- FN are false negatives.

Moreover, deep learning has been used in extracting complex statistics and learning high level features from big data. However, it may suffer from poor performance when learning from datasets with limited availability. To overcome the challenges, Google DeepMind [32] applied a few-shot learning approach, which is based on meta-learning and memory-augmented neural networks (MANNs) requiring only a few samples (e.g., 1, 5, or 10) per class for training. In the experiment for Omniglot classification, the MANN outperformed the human participants, and long short-term memory (LSTM), reaching up to 94.9% and 98.1% accuracy by the 5th and 10th instances, respectively. In general, few-shot learning [36] is a type of machine lerning problem, which aims to learn from a limited number of examples in experience with supervised information for some classes of trask. It popularity has grown. For instance, Snell et al. [33] used the FSL to achieve 68.20% accuracy on the miniImageNet dataset with only 5-shot modelling and an impressive 99.7% (5-way) accuracy with 5 shots on the Omniglot dataset. Their networks perform especially well by using an episodic training model. They generalized these results to zero-shot learning. As a preview, in this paper, we explore an alternative approach that trains over a small set of data, and we aim to achieve a better accuracy.

3 OUR DATA SCIENCE SOLUTION

Recall from Section 2 that autoencoder architecture can be used for learning internal representations by error propagation. Here, we adapt the autoencoder architecture to create a similar solution for the reconstruction of our input. Our solution generates multiple outputs:

- the reconstructed input, and
- the output for cancer prediction.

We also use multitask learning. With it, the encoded features in our solution contain more information on the cell nucleus. Moreover, we propose two algorithms to predict cancerous cell on a few data samples. To predict the learning representation of the input features, the algorithms project the learned features into a compact encoded feature vector in the middle layer (destinated to learn from auxiliary tasks) of the network layers. Let us consider the whole cancer data sample with both the input and the labels as

$$\{(c_i, y_i)\}_{i=1}^{n}$$

where:

- n is the total number of data samples,
- c_i is an input attribute feature for the cancer dataset, and
- y_i is a label that contains either cancerous or non-cancerous.

Here, we assume that there is only a limited number of samples by representing the few data samples as

$$\{F_i\}_{i=1}^{m}$$

where m is the total number of limited data samples with labels such that $m \in \{1, 5, 10\}$ and $m \ll n$.

3.1 Our First Healthcase Predictive Analytics (HPA1) Algorithm

Figure 1 shows an overview of the architecture of our data science framework. In it,

- the green layers represent the encoder layers,
- the blue layers represent the decoder layers,
- the combination of these green and blue layers form an autoencoder,
- the orange layers represent the cancer prediction output, and
- all layers are created using fully connected neurons.

Hence, the reconstruction features for our HPA1 can be represented as an autoencoder with the combination of the green and the blue layers.

Our first healthcare predictive analytics (HPA1) algorithm consists of training the autoencoder and the few labeled data samples *simultaneously*. The training for the network occurs by

- reconstructing the original input features as an auxiliary task, and
- predicting the true value for the cancerous output.

More precisely, our HPA1 can be represented by:

- the reconstruction features as $\{R_i\}_{i=1}^{n}$;
- the cancer output prediction as y, which is the ground truth for the cancer prediction with $y \in \{0, 1\}$;
- the encoded features as $\{E_i\}_{i=1}^{n}$; and
- the features fed into the autoencoder as $\{F_i\}_{i=1}^{n}$.

Here, the loss function L_R for the reconstruction of the original input is a mean absolute error:

$$L_R = \frac{1}{N} \sum_{i=1}^{N} \left| F_i - \widehat{F_i} \right| \quad (6)$$

where

- N is the number of trained data samples, and
- $\widehat{F}$ is the reconstructed features of the cancer cells.

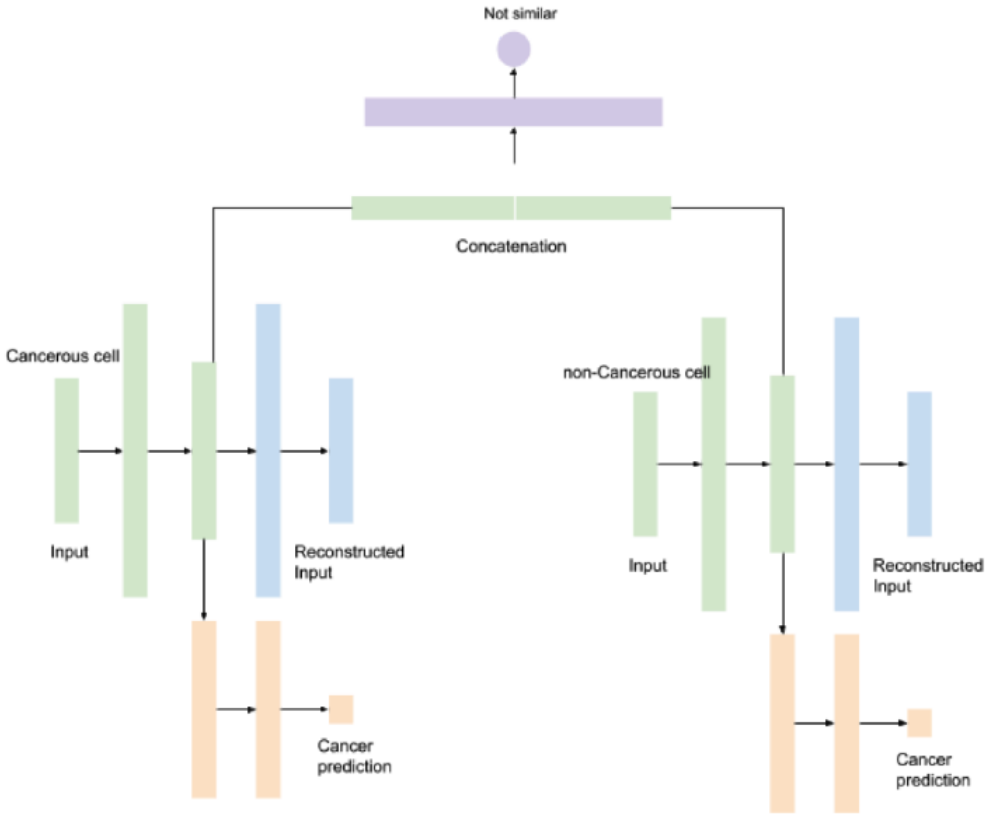


Figure 1: Network of our proposed architecture.

As for the loss function L_O for the cancer prediction output, we use the following binary cross-entropy loss:

$$L_O = \frac{1}{M}\sum_{i=1}^{M}(y_i \log(y_i) + (1-y_i)\log(1-y_i)) \tag{7}$$

The total loss function L_{OR} for our HPA1 algorithm is:

$$L_{OR} = L_O + \lambda L_R \tag{8}$$

where λ is set to 0.5 for our experiment as we want to emphasize the cancer prediction loss when training simultaneously so that the network will generalize more on the importance of predicting the label of cancer correctly rather than putting higher priority in reconstructing the input. We train our network for 20 epochs with a batch size of 32 and an optimizer to backpropagate the gradients for our network with a learning rate of 0.001.

3.2 Our Second Healthcase Predictive Analytics (HPA2) Algorithm

Our second healthcare predictive analytics (HPA2) algorithm consists of training the autoencoder *before* training the cancer prediction output. Here, after HPA2 trained the encoder layers through the reconstruction of the input, the algorithm freezes the encoder layers and uses the encoded features (instead of the original attribute features) for training to predict the cancer output. The network receives the original attribute features and encodes the original attribute features into a compact encoded feature. The compact encoded feature then passes into multiple fully connected layers to predict the cancer output.

The loss function L_R for this HPA2 algorithm is similar to our HPA1 algorithm. The reconstruction loss for the autoencoder is as follows:

$$L_R = \frac{1}{N}\sum_{i=1}^{N}\left|F_i - \widehat{F_i}\right| \tag{9}$$

where

- N is the number of trained data samples,
- $\{F_i\}_{i=1}^{n}$ is the features fed into the autoencoder, and
- $\left\{\widehat{F_i}\right\}_{i=1}^{n}$ is the reconstructed features by the autoencoder.

The reconstruction phase is trained for 100 epochs by using an optimizer with a learning rate of 0.001.

After training of the network using the reconstruction loss, the encoder layer is frozen to prevent the weights from changing in the next training phase. The next training phase includes the training of the cancer prediction through the encoded features that is trained from the reconstruction phase. The loss function L_O for the cancer

prediction through the encoded features is:

$$L_O = \frac{1}{M}\sum_{i=1}^{M}(y_i \log(y_i) + (1-y_i)\log(1-y_i)) \quad (10)$$

Similarly, the cancer prediction loss is trained for 20 epochs by using an an optimizer with a learning rate of 0.001.

3.3 Network Architecture of Our Data Science Solution Framework

Our network architecture in our data science solution framework consists of

(1) an autoencoder, and
(2) a cancer prediction module.

The *autoencoder* consists of four fully connected hidden layers to reconstruct the input:

- The first hidden layer contains 64 neurons.
- the second hidden layer contains 16 neurons and the encoded features for training the other networks (e.g., cancer prediction).
- The third hidden layer contains 64 neurons.
- The last layer is the output layer containing the same total number of features as the input features.

Our autoencoder architecture has the first layer and the last hidden layer inflated to produce a more compact and structured representation of the input in the encoded feature layer.

The *cancer prediction module* of our network consists of three fully connected layers:

- The first fully connected layer contains eight neurons.
- The second fully connected layer contains eight neurons.
- The third fully connected layer, which is also the output layer, contains a single neuron to predict if the input received is cancerous or not cancerous.

All the hidden layers contain a *rectified linear unit* (*ReLU*) activation. Our two algorithms HPA1 and HPA2 use both the autoencoder module and the cancer prediction module.

4 EVALUATION

4.1 Real-Life Datasets for Evaluation

To evaluate our data science solution framework with its two healthcare predictive analytics algorithms HPA1 and HPA2, we conducted experiments on three open datasets on breast cancer from UC Irvine Machine Learning Repository [15] as mentioned in Section 2:

(1) Breast cancer Wisconsin (original) data set (aka original *Wisconsin breast cancer database* (*WBC*)): It captures 699 breast cancer samples or instances.
(2) Breast cancer Wisconsin (diagnostic) data set (aka *Wisconsin diagnostic breast cancer database* (*WDBC*)): It captures 569 breast cancer instances, and each with 10 features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass describing characteristics of the cell nuclei present in the image.
(3) Breast cancer Wisconsin (prognostic) data set (aka *Wisconsin prognostic breast cancer database* (*WPBC*)): It captures 198 breast cancer cases, and each case represent consecutive follow-up data for a patient who exhibited invasive breast cancer and no evidence of distant metastases at the time of diagnosis.

4.1.1 Wisconsin Breast Cancer Database (*WBC*). The WBC dataset captures 699 breast cancer samples or instances. These samples arrived periodically as a particular university hospital physician in Wisconsin [28, 37] reported his clinical cases during 1989-1991. Hence, the database reflects this chronological grouping of the data. Each instance consists of (1) a sample code identifier, (2-10) 9 useful features with integer values, and (11) a class label, for a total of 11 attributes:

(1) Sample code number: ID number
(2) Clump thickness: 1-10
(3) Uniformity of cell size: 1-10
(4) Uniformity of cell shape: 1-10
(5) Marginal adhesion: 1-10
(6) Single epithelial cell size: 1-10
(7) Bare nuclei: 1-10
(8) Bland chromatin: 1-10
(9) Normal nucleoli: 1-10
(10) Mitoses: 1-10
(11) Class: 2 for benign, 4 for malignant

See Table 1 for a sample of this WBC dataset.

Among 699 instances, 16 of them contain a single missing (i.e., univariate) attribute value denoted by "?". With this dataset, we train our data science solution trains the cancer prediction module with the 9 useful features with a goal to predict if the cell is malignant or benign. In terms of data distribution, 458 instances (i.e., 65.5% of all 699 instances) were benign (i.e., not medicially harmful) and the remaining 241 instances (i.e., 34.5%) were malignant (i.e., cancer). Figure 2 visualizes this WBC dataset after applying dimension reduction via the principal component analysis (PCA). The blue data points represent the benign (i.e., non-cancerous) samples, and the red data points represent the malignant (i.e., cancerous) samples.

4.1.2 Wisconsin Diagnostic Breast Cancer Database (*WDBC*). The WDBC dataset captures 569 breast cancer instances, and each with 10 features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass describing characteristics of the cell nuclei present in the image. Each instance consists of (1) an identifier, (2) diagnosis, as well as (3-32) mean, standard error (SE) and "worst" (or largest) of 10 useful features with real-number values, for a total of 32 attributes: The real valued features for both the Wisconsin Diagnostic Breast Cancer and Wisconsin Prognostic Breast Cancer are as follows:

(1) ID number
(2) Diagnosis: M = malignant, B = benign
(3) Mean radius (mean of distances from center to points on the perimeter)
(4) Mean texture (standard deviation of gray-scale values)
(5) Mean perimeter
(6) Mean area
(7) Mean smoothness (local variation in radius lengths)
(8) Mean compactness $\left(= \frac{\text{perimeter}^2}{\text{area}} - 1.0\right)$

Table 1: Samples of the WBC dataset.

Code number	Clump thickness	Uniformity cell shape	Uniformity cell size	Marginal adhesion	Single epithelial cell size	Bare nuclei	Bland chromatin	Normal nucleoli	Mitoses	Class botline
1000025	5	1	1	1	2	1	3	1	1	2
1002945	5	4	4	5	7	10	3	2	1	2
1015425	3	1	1	1	2	2	3	1	1	2

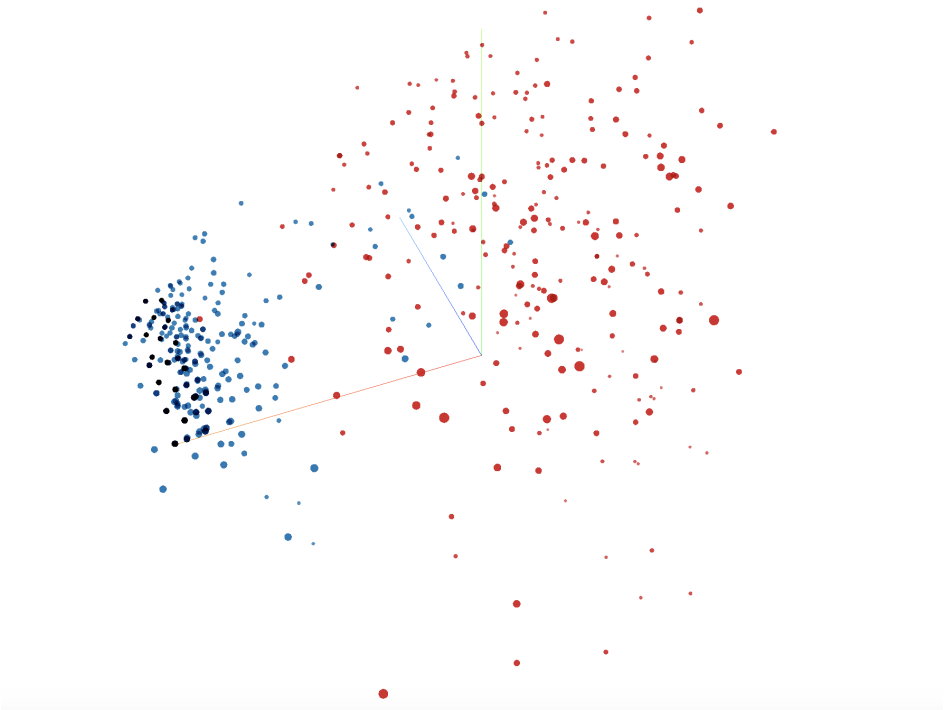


Figure 2: Visualization of the WBC dataset with 9 dimensions reduced by PCA showing 458 benign instances as blue data points and 241 malignant instances as red data points.

(9) Mean concavity (severity of concave portions of the contour)
(10) Mean concave points (number of concave portions of the contour)
(11) Mean symmetry
(12) Mean fractal dimension (= coastline approximation −1)
(13) Radius standard error (SE)
(14) Texture SE
(15) Perimeter SE
(16) Area SE
(17) Smoothness SE
(18) Compactness SE
(19) Concavity SE
(20) Concave point SE
(21) Symmetry SE
(22) Fractal dimension SE
(23) "Worst" radius
(24) "Worst" texture
(25) "Worst" perimeter
(26) "Worst" area
(27) "Worst" smoothness
(28) "Worst" compactness
(29) "Worst" concavity
(30) "Worst" concave points
(31) "Worst" symmetry
(32) "Worst" fractal dimension

Unlike the WBC dataset, there is no missing value in this WDBC dataset. With this dataset, we train our data science solution trains the cancer prediction module with the 30 useful features with a goal to predict if the cell is malignant or benign. In terms of data distribution, 357 instances (i.e., 62.7% of all 569 instances) were benign (i.e., not medicially harmful) and the remaining 212 instances (i.e., 37.3%) were malignant (i.e., cancer). Figure 3 visualizes this WBC dataset after applying dimension reduction via the principal component analysis (PCA). The blue data points represent the benign

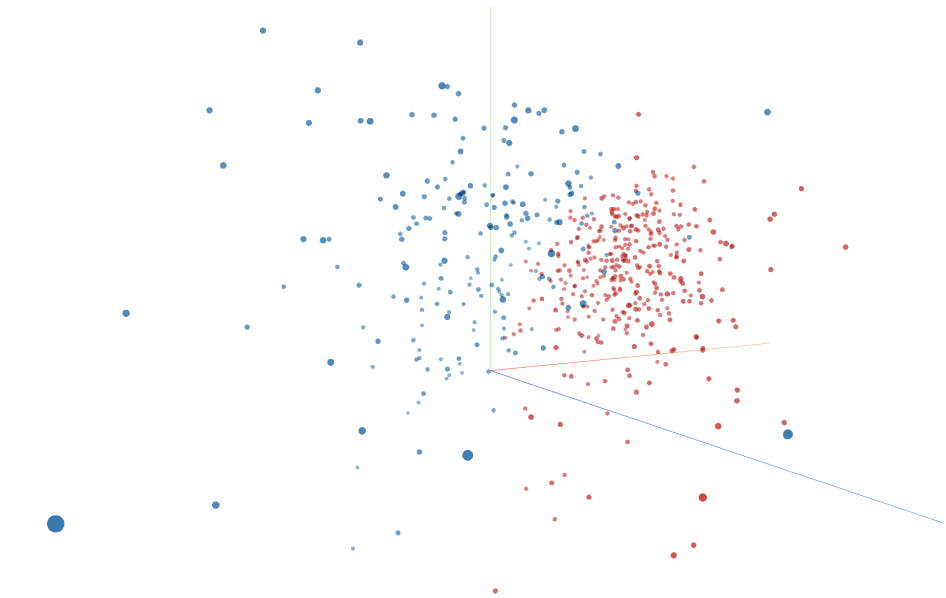


Figure 3: Visualization of the WDBC dataset with 30 dimensions reduced by PCA showing 357 benign instances as blue data points and 212 malignant instances as red data points.

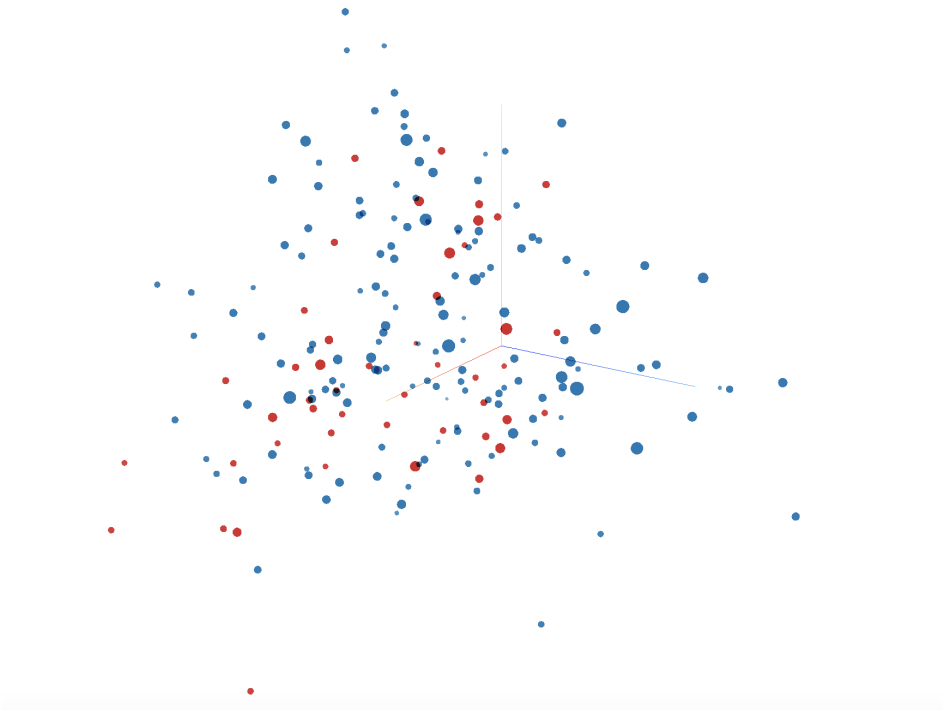


Figure 4: Visualization of the WPBC dataset with 32 dimensions reduced by PCA showing 47 recurrent cases as blue data points and 151 non-recurrent cases as red data points.

(i.e., non-cancerous) samples, and the red data points represent the malignant (i.e., cancerous) samples.

4.1.3 Wisconsin Prognostic Breast Cancer Database (WPBC). The WPBC dataset captures 198 breast cancer cases, and each case represent consecutive follow-up data for a patient who exhibited invasive breast cancer and no evidence of distant metastases at the time of diagnosis. Like the WBDC dataset, each instance in this WPBC dataset contains 10 features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass describing characteristics of the cell nuclei present in the image. Each instance consists of (1) an identifier, (2) outcome, (3) recurrence or disease free time, (4-33) mean, standard error (SE) and "worst" (or largest) of 10 useful features with real-number values, (34) tumor size, and (35) lymph node status, for a total of 35 attributes:

(1) ID number
(2) Outcome: R = recurrent, N = non-recurrent
(3) Time: recurrence time (for recurrent outcome), disease-free time (for non-recurrent outcome)
(4-33) (*identical to attributes (3-32) in the WDBC dataset*)
(34) Tumor size (diameter of the excised tumor in cm)
(35) Lymph node status (number of positive axillary lymph nodes observed at time of surgery)

Unlike the WDBC dataset (with no missing values), lymph node status was missing from 4 of the 198 cases. With this dataset, we train our data science solution trains the cancer prediction module with the 32 useful features with a goal to predict if the case is recurrent before 24 months (i.e., 2-year recurrence) or not. In terms of data distribution, 151 cases (i.e., 76.3% of all 198 cases) were non-recurrent and the remaining 47 cases (i.e., 23.7%) were recurrent. Figure 4 visualizes this WPBC dataset after applying dimension reduction via the principal component analysis (PCA). The blue data points represent the benign (i.e., non-cancerous) samples, and the red data points represent the malignant (i.e., cancerous) samples.

4.2 Related Models for Comparisons

For evaluation, we compared our data science solution framework with its two healthcare predictive analytics algorithms HPA1 and HPA2 with several existing machine learning models including the following:

- random forest,
- gradient boosting,
- support vector classification (SVC), and
- linear SVC.

We ran these models on the three datasets described in Section 4.1 with their default parameters.

4.3 Evaluation Process

We first preprocessed the three breast cancer datasets by performing a standard scaler normalization to normalize the provided data samples. Afterwards, we tested the models on different number of labeled samples {1, 5, 10}. For the WBC dataset, the total number of classes is two—namely, cancerous and non-cancerous. Hence, we conducted our experiments with a total number of limited samples—namely, {2, 10, 20} samples—from the WBC datasets, where {1, 5, 10} of the limited number of labeled data samples for each class was given. We used the remaining unused data samples to train the autoencoder for the reconstruction of the data samples. We set 20 data samples for each class as the test samples to determine if our solution is overfit and used the remaining data samples as validation set. We applied an optimizer [18] to our solution and set the learning rate to 0.001. Our batch size was set to 32.

For our HPA1 algorithm, as the reconstruction and the cancer prediction are trained simultaneously, we ran training on both the reconstruction and cancer prediction with 50 epochs (where early stopping may occur when the model gets overfit throughout the epochs). The cancer prediction was trained only on the limited labeled data samples, whereas the reconstruction accessed all the data samples without the labels. Since reconstruction only needs the inputs but not labels, we send all the unlabeled data samples to the autoencoder module for training.

For our HPA2 algorithm, the reconstruction and the cancer prediction were trained separately. The reconstruction of the data samples was first trained with 150 epochs. The reconstruction phase accessed all the data samples without the labels. After the reconstruction phase was completed, the autoencoder module was frozen to prevent re-training of the encoded features obtained from training the network during the reconstruction phase. The cancer prediction module used the encoded features to predict on the cancer data samples. Only the limited labeled data samples were provided for training our cancer prediction module. During training in the cancer prediction phase, the only network layers that were trained were the orange layers as shown in Figure 1. The cancer prediction module was trained with 50 epochs.

4.4 Evaluation Results

Table 2 shows the F1 score for the machine learning models on the WBC dataset when training on {1, 5} of the labeled data samples for each class. Each experiment was ran for 5-fold cross validation and the average F1 score was computed by Eq. (5). Results show that our data science solution framework with the HPA1 algorithm led to the highest F1 scores when compared with the other four related machine learning models.

Table 2: Evaluation results (F1 scores) on the WBC dataset (with the highest score of each column highlighted in bold).

#samples per class	1	5
Random forest	0.79	0.89
Gradient boost	0.70	0.89
SVC	0.70	0.95
Linear SVC	0.75	0.92
Our HPA1	**0.80**	**0.95**

Similarly, Table 3 shows the F1 score for the machine learning models on the WBC dataset when training on {1, 5} of the labeled data samples for each class. Again, each experiment was ran for 5-fold cross validation and the average F1 score was computed by Eq. (5). Results show that our data science solution framework with the HPA1 algorithm led to the highest F1 scores when compared with the other four related machine learning models.

Table 3: Evaluation results (F1 scores) on the WDBC dataset (with the highest score of each column highlighted in bold).

#samples per class	1	5
Random forest	0.86	0.88
Gradient boost	0.85	0.88
SVC	0.86	0.90
Linear SVC	0.895	0.910
Our HPA1	**0.896**	**0.911**

Table 4 shows the F1 score for the machine learning models on the WPBC dataset when training on {1, 5, 10} of the labeled data samples for each class. Again, each experiment was ran for 5-fold cross validation and the average F1 score was computed by Eq. (5). Results show that our data science solution framework with the HPA1 and HPA2 algorithms led to the highest F1 scores when compared with the other four related machine learning models.

Table 4: Evaluation results (F1 scores) on the WBC dataset (with the highest score of each column highlighted in bold).

#samples per class	1	5	10
Random forest	0.42	0.46	0.54
Gradient boost	0.35	0.40	0.43
SVC	0.33	0.46	0.45
Linear SVC	0.51	0.50	0.49
Our HPA1	0.62	0.64	0.70
Our HPA2	**0.79**	**0.81**	**0.79**

Observed from Figures 2 and 3, both the WBC and WDBC datasets have a clear separation between the cancerous and non-cancerous cells. This contributes to the relatively well performance by related baseline machine learning models, though our HPA1 algorithms performed even better, on the two datasets as shown in Tables 2 and tab3. However, the situation is different here for the WPBC dataset, which contains a more complex data samples. Observed from Figure 4, the recurring and non-recurring cancer cells are concentrated and crossed over with each other. This explains why our network data science solution framework with the HPA1 and HPA2 algorithms outperformed the other four related machine learning models. To elaborate, the baseline machine learning models all led to F1 scores of at most 0.54, whereas our HPA1 algorithm led to much higher F1 scores of at least 0.62 to 0.70 (i.e., 0.08 to 0.37 higher in F1 scores than the baseline models). The F1 scores for our HPA2 algorithms were even higher, with at least 0.79 to 0.81 (i.e., 0.25 to 0.48 higher in F1 scores than the baseline models). The results show that our solution provide more accurate prediction. The results also illustrate the benefits of our solution, especially for complex data when data points are clustered (e.g., concentrated and crossed over with each other).

Moreover our HPA2 algorithm performed much better than our HPA1 algorithm because the latter learns both the reconstruction and the cancer prediction output simultaneously (which does not give the network a non-interrupting training) to create an encoded feature that solely based on reconstruction of the input. In constrast, our HPA2 algorithm trained for the reconstruction first before training the cancer prediction module (which creates an uninterrupted training for the encoded features through reconstruction). As the autoencoder layers is frozen after reconstruction is fully trained, the cancer prediction module is able to learn to predict cancerous cell through the encoded features without re-training the autoencoder layers.

5 CONCLUSIONS

In this paper, we presented a data science solution framework with two healthcare predictive analytics algorithms HPA1 and HPA2. We demonstrated our solution by conducting evaluation on three real-life open datasets about breast cancer. The evaluation results show that our solution does not require lots of training data, which is benefits to healthcare and medicial fields when data are expensive to obtain. The results also show that our solution outperforms many existing machine learning models, especially when data are complex. It is important to note that, although we conducted evaluation on breast cancer data, the solution is expected to be applicable to many other healthcare predictive analytics tasks (e.g., prediction on lung cancers). As ongoing and future work, we transfer knowledge learned here and adapt our solution to other healthcare predictive analytics domains (e.g., from perdicting outcomes to predicting recurrent time). Moreover, we also explore possibility to incorporate other techniques [5, 8, 10–12] (e.g., visual analytics [23]) into our tool to further enhance data science for healthcare predictive analytics.

ACKNOWLEDGMENTS

This project is partially supported by NSERC (Canada) and University of Manitoba.

REFERENCES

[1] P. Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. JMLR 27, pp. 37-50.
[2] S.D. Bernhard, C.K. Leung, V.J. Reimer, J.Westlake. 2016. Clickstream prediction using sequential stream mining techniques with Markov chains. IDEAS 2016, pp. 24-33.
[3] P. Braun, A. Cuzzocrea, C.K. Leung, A.G.M. Pazdor, S.K. Tanbeer, G.M. Grasso. 2018. An innovative framework for supporting frequent pattern mining problems in iot environments. ICCSA, Part V, pp. 642-657.
[4] R. Caruana. 1997. Multitask learning. Machine Learning 28(1), pp. 41-75.
[5] G. Chatzimilioudis, A. Cuzzocrea, D. Gunopulos, N. Mamoulis. 2013. A novel distributed framework for optimizing query routing trees in wireless sensor networks via optimal operator placement. JCSS 79(3), pp. 349-368.
[6] J.P. Choi, T.H. Han, R.W. Park. 2009. A hybrid Bayesian network model for predicting breast cancer prognosis. J. Korean Soc. Med. Inf. 15(1), pp. 49-57.
[7] D. Choudhery, C.K. Leung. 2017. Social media mining: prediction of box office revenue. IDEAS 2017, pp. 20-29.
[8] A. Cuzzocrea, E. Bertino. 2011. Privacy preserving OLAP over distributed XML data: a theoretically-sound secure-multiparty-computation approach. JCSS 77(6), pp. 965-987.
[9] A. Cuzzocrea, G.M. Grasso, F. Jiang, C.K. Leung. 2016. Mining uplink-downlink user association in wireless heterogeneous networks. IDEAL 2016, pp. 533-541.
[10] A. Cuzzocrea, R. Moussa, G. Xu. 2013. OLAP*: effectively and efficiently supporting parallel OLAP over big data. MEDI 2013, pp. 38-49.
[11] A. Cuzzocrea, C. Mastroianni, G.M. Grasso. 2016. Private databases on the cloud: models, issues and research perspectives. BigData 2016, pp.3656-3661.
[12] A. Cuzzocrea, V. Russo. 2013. Privacy preserving OLAP and OLAP security. Encyclopedia of Data Warehousing and Mining 2009, pp. 1575-1581.
[13] A. Demiraj, K. Karozos, I. Spartalis, V. Vassalos. 2019. Meta-data management and quality control for the medical informatics platform. IDEAS 2019, 10:1-10:9.

[14] S. Dolgikh. 2018. Spontaneous concept learning with deep autoencoder, Int. J. Comp. Intel. Syst. 12(1), pp. 1-12.
[15] D. Dua, C. Graff. 2019. UCI machine learning repository. http://archive.ics.uci.edu/ml
[16] A. Fariha, C.F. Ahmed, C.K. Leung, S.M. Abdullah, L. Cao. 2013. Mining frequent patterns from human interactions in meetings using directed acyclic graphs. PAKDD 2013, Part I, pp. 38-49.
[17] F. Jiang, C.K. Leung, S.K. Tanbeer. 2012. Finding popular friends in social networks. CGC 2012, pp. 501-508.
[18] D.P. Kingma, J. Ba. 2014. Adam: a method for stochastic optimization. arXiv:1412.6980
[19] M.A. Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. AIChE J. 37(2), pp. 233–243.
[20] C.K. Leung. 2014. Uncertain frequent pattern mining. Frequent Pattern Mining, pp. 417-453.
[21] C.K. Leung. 2018. Frequent itemset mining with constraints. Encyclopedia of Database Systems, 2nd edn, pp. 1531-1536.
[22] C.K. Leung. 2019. Pattern mining for knowledge discovery. IDEAS 2019, pp. 34:1-34:5.
[23] C.K. Leung, C.L. Carmichael. 2009. FpVAT: a visual analytic tool for supporting frequent pattern mining. ACM SIGKDD Explor. 11(2), pp. 39-48.
[24] C.K. Leung, R.K. MacKinnon, Y. Wang. 2014. A machine learning approach for stock price prediction. IDEAS 2014, pp. 274-277.
[25] C.K. Leung, M.A.F. Mateo, D.A. Brajczuk. 2008. A tree-based approach for frequent pattern mining from uncertain data. PAKDD 2008, pp. 653-661.
[26] C.K. Leung, S.K. Tanbeer, J.J. Cameron. 2014. Interactive discovery of influential friends from social networks. SNAM 4(1), pp. 154:1-154:13.
[27] C.K. Leung, Y.B. Zhang, C. S.H. Hoi, J. Souza, B. H. Wodi. 2019. Big data analysis and services: visualization of smart data to support healthcare analytics. IEEE SmartData 2019, pp. 1261-1268.
[28] O.L. Mangasarian, W.N. Street, W.H. Wolberg. 1995. Breast cancer diagnosis and prognosis via linear programming. Oper. Res. 43(4), pp. 570-577.
[29] I. Pradhan, K. Potika, M. Eirinaki, P. Potikas. 2019. Exploratory data analysis and crime prediction for smart cities. IDEAS 2019, pp. 4:1-4:9.
[30] K.U. Rani. 2010. Parallel approach for diagnosis of breast cancer using neural network technique, IJCA 10(3), pp. 1-5.
[31] D.E. Rumelhart, G.E. Hinton, R.J. Williams. 1986. Learning internal representations by error propagation. Parallel Distributed Processing, Vol 1, pp. 318-362.
[32] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap. 2016. One-shot learning with memory-augmented neural networks. arXiv:1605.06065.
[33] J. Snell, K. Swersky, R. Zemel. 2017. Prototypical networks for few-shot learning, arXiv:1703.05175
[34] J. Souza, C.K. Leung, A. Cuzzocrea. 2020. An innovative big data predictive analytics framework over hybrid big data sources with an application for disease analytics. AINA 2020, pp. 669-680.
[35] S.K. Tanbeer, C.K. Leung, J.J. Cameron. 2014. Interactive mining of strong friends from social networks and its applications in e-commerce. JOCEC 24(2-3), pp. 157-173.
[36] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni. 2020. Generalizing from a few examples: a survey on few-shot learning. ACM CSur 53(3), pp. 61:1-63:34.
[37] W.H. Wolberg, O.L. Mangasarian. 1990. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc Natl Acad Sci USA 87(23), pp 9193-9196.
[38] M.C.S. Wong, W.B. Goggins, H.H.X. Wang, F.D.H. Fung, C. Leung, S.Y.S. Wong, C.F. Ng, J.J.Y. Sung. 2016. Global incidence and mortality for prostate cancer: analysis of temporal patterns and trends in 36 countries. Eur. Urol. 70(5), pp. 862-874.
[39] G. Zhang, Y. Liu, X. Jin. 2020. A survey of autoencoder-based recommender systems. FCSc 14(2), pp. 430-450.
[40] K. Zhang, L. Wu, Z. Zhu, J. Deng. 2020. A multitask learning model for traffic flow and speed forecasting. IEEE Access 8, pp. 80707-80715.
[41] Q. Zhou, B. Yong, Q. Lv, J. Shen, X. Wang. 2020. Deep autoencoder for mass spectrometry feature learning and cancer detection. IEEE Access 8, pp. 45156-45166.

A Novel Spatio-Temporal Interpolation Algorithm and Its Application to the COVID-19 Pandemic

Junzhe Cai
University of Nebraska-Lincoln
Lincoln, Nebraska
jcai@huskers.unl.edu

Peter Z. Revesz
University of Nebraska-Lincoln
Lincoln, Nebraska
revesz@cse.unl.edu

ABSTRACT

This paper describes several interpolation methods for predicting the number of cases of the COVID-19 pandemic. The interpolation methods include some well-known temporal interpolation algorithms including Lagrange interpolation, cubic spline interpolation, and exponential decay interpolation. These temporal interpolation algorithms enable the interpolation of the COVID-19 cases at locations where measures on prior days are available. However, pandemics are not purely temporal but spatio-temporal phenomena. Therefore, the neighboring locations need to be considered too in order to derive accurate interpolation values for future days. This paper introduces a novel spatio-temporal interpolation algorithm that is shown to be better than any purely temporal interpolation algorithm in predicting the COVID-19 cases in the continental United States. In particular, the novel spatio-temporal interpolation method achieves a mean absolute error of 8.44 cases over a million people when predicting two days ahead the number of cases of the COVID-19 pandemic.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; *Machine learning approaches*; • **Information Systems** → *Data mining*; • **Mathematics of computing** → **Interpolation**.

KEYWORDS

COVID-19, exponential decay, interpolation, inverse distance weighting, Langrange, Long Short-Term memory, prediction, Recurrent Neural Network, spatio-temporal, temporal

ACM Reference Format:
Junzhe Cai and Peter Z. Revesz. 2020. A Novel Spatio-Temporal Interpolation Algorithm and Its Application to the COVID-19 Pandemic. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410602

1 INTRODUCTION

In many applications, the value of a spatio-temporal variable needs to be predicted for some time in the future based on previously measured data at the same location and neighboring locations. Some well-known applications of these include the prediction of economic indicators, such as stock prices, GDP or unemployment figures. In this paper, we take a look at predicting the number of cases of the COVID-19 pandemic [5], which is a novel type of pandemic with no well-tested prediction algorithms for it. Earlier epidemics prediction algorithms exist but they often require extra information like infected animals that are not available or applicable in this case [10]. Therefore, we focus on the COVID-19 pandemic in this paper, although our novel spatio-temporal interpolation algorithm may also be applicable to other spatio-temporal interpolation problems [8].

The rest of this paper is organized as follows. Section 2 reviews previous temporal and spatial interpolation algorithms and basic concepts. Section 3 describes a novel spatio-temporal interpolation algorithm that we developed to predict the number of cases of the COVID-19 pandemic. Section 4 presents the experimental results. Finally, Section 5 gives some conclusions and outlines future work.

2 REVIEW OF PREVIOUS INTERPOLATION ALGORITHMS AND BASIC CONCEPTS

2.1 Temporal Interpolation Methods

Let at some location y_i be the number of cases of the COVID-19 pandemic i days ago. Hence y_1 is the number of cases yesterday, and y_2 is the number of cases the day before yesterday etc. Then the *Best Fit Cubic* and the *Lagrange* interpolation methods [1] can be used to predict the number of cases of the COVID-19 pandemic at that location. These methods derive interpolation functions into which we can place any future time instance to get a prediction value. In addition, the exponential decay temporal method can be used to get an estimate for the current day using the following formula, which assumes that we know the number of cases during the six previous days:

$$y = \frac{1}{2}y_1 + \frac{1}{4}y_2 + \frac{1}{8}y_3 + \frac{1}{16}y_4 + \frac{1}{32}y_5 + \frac{1}{32}y_6 \qquad (1)$$

The above formula can be extended for more numbers of days. The important feature is that the weights are successively diminishing by half except in the last instance, where the last weight is equal to the previous weight. Note that in this way, the sum of all the weights is exactly one. Finally, another prediction method that was proposed by Revesz [9] uses the following formula to predict the number of cases of the COVID-19 pandemic, where t is the number of days ahead from the last data. In other words, if the last data is for yesterday, then predicting for today means $t = 1$ and for tomorrow $t = 2$ etc.

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410602

$$y = \frac{t^2}{2}y_3 + (t + t^2)y_2 + (1 + t + \frac{t^2}{2})y_1 \quad (2)$$

2.2 Spatial Interpolation Methods

Inverse Distance Weighting (IDW) [13] is a common spatial interpolation method. It is used when the interpolated variable at a location has a weighted relationship with its neighbors and when that relationship varies with distance. If a neighbor is closer than another neighbor, then the weight of the former will be higher than the weight of the latter. We use λ_i as the weight for every neighbor and y_i as the interpolated variable for every neighbor. Then the Inverse Distance Weighting equation for the interpolated variable y at a location can be written in terms of its neighbors as follows:

$$y = \sum_{i=1}^{N} \lambda_i \times y_i \quad (3)$$

where the equation for calculating λ_i can be written as follows:

$$\lambda_i = \frac{(1/d_i)^P}{\sum_{k=1}^{N}(1/d_i)^P} \quad (4)$$

The p value can be any number ≥ 1. For simplicity, in this paper we assume that p = 1.

2.3 Moving Average

In order to have smoother data, the moving average is applied. Rather than use the data for a single day, we use the moving average value for five days. For example, in the state of Alabama in the USA, the number of COVID-19 cases for the days from May 20 to May 24 were the following in order: 676, 362, 256, 479, and 329. Hence the five day moving average centered on May 22nd is the average of these five values, that is, 420.4. This explains the last value in the first row of Table 1 that starts with Alabama.

2.4 Error Measures

To experimentally evaluate the accuracy of the interpolation methods, we use the *Mean Absolute Error* (MAE) and the *Root Mean Square Error* (RMSE) measures, which are defined as follows, where F_i is the predicted value and A_i is the corresponding actual value and N is the number of items:

$$MAE = \frac{\sum_{i=1}^{N} |F_i - A_i|}{N} \quad (5)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(F_i - A_i)^2}{N}} \quad (6)$$

2.5 Recurrent Neural Networks

RNN (Recurrent Neural Networks) was developed based on David et al. [11] in 1988. RNN [4] contains the ability of time series prediction, motor control in non-Markovian environment and rhythm detection. Figure 1 shows the architecture of the RNN, which differs from other neural networks in that RNN contains one or more than one loop between nodes. RNN has a limit when dealing with back-propagated error. One of the extensions of RNN called LSTM (Long

Table 1: Latitude and longitude of the continental US states.

State	Latitude	Longitude
Alabama	32.31823	-86.902298
Arizona	34.048927	-111.093735
Arkansas	34.799999	-92.199997
California	36.116203	-119.681564
Colorado	39.059811	-105.311104
Connecticut	41.597782	-72.755371
Delaware	39.318523	-75.507141
Florida	27.766279	-81.686783
Georgia	33.040619	-83.643074
Idaho	44.240459	-114.478828
Illinois	40.349457	-88.986137
Indiana	39.849426	-86.258278
Iowa	42.011539	-93.210526
Kansas	38.5266	-96.726486
Kentucky	37.66814	-84.670067
Louisiana	31.169546	-91.867805
Maine	44.693947	-69.381927
Maryland	39.063946	-76.802101
Massachusetts	42.230171	-71.530106
Michigan	43.326618	-84.536095
Minnesota	45.694454	-93.900192
Mississippi	32.741646	-89.678696
Missouri	38.456085	-92.288368
Montana	46.921925	-110.454353
Nebraska	41.12537	-98.268082
Nevada	38.313515	-117.055374
New Hampshire	43.452492	-71.563896
New Jersey	40.298904	-74.521011
New Mexico	34.840515	-106.248482
New York	42.165726	-74.948051
North Carolina	35.630066	-79.806419
North Dakota	47.528912	-99.784012
Ohio	40.388783	-82.764915
Oklahoma	35.565342	-96.928917
Oregon	44.572021	-122.070938
Pennsylvania	40.590752	-77.209755
Rhode Island	41.680893	-71.51178
South Carolina	33.856892	-80.945007
South Dakota	44.299782	-99.438828
Tennessee	35.747845	-86.692345
Texas	31.054487	-97.563461
Utah	40.150032	-111.862434
Vermont	44.045876	-72.710686
Virginia	37.769337	-78.169968
Washington	47.400902	-121.490494
West Virginia	38.491226	-80.954453
Wisconsin	44.5	-89.5
Wyoming	43.07597	-107.290283

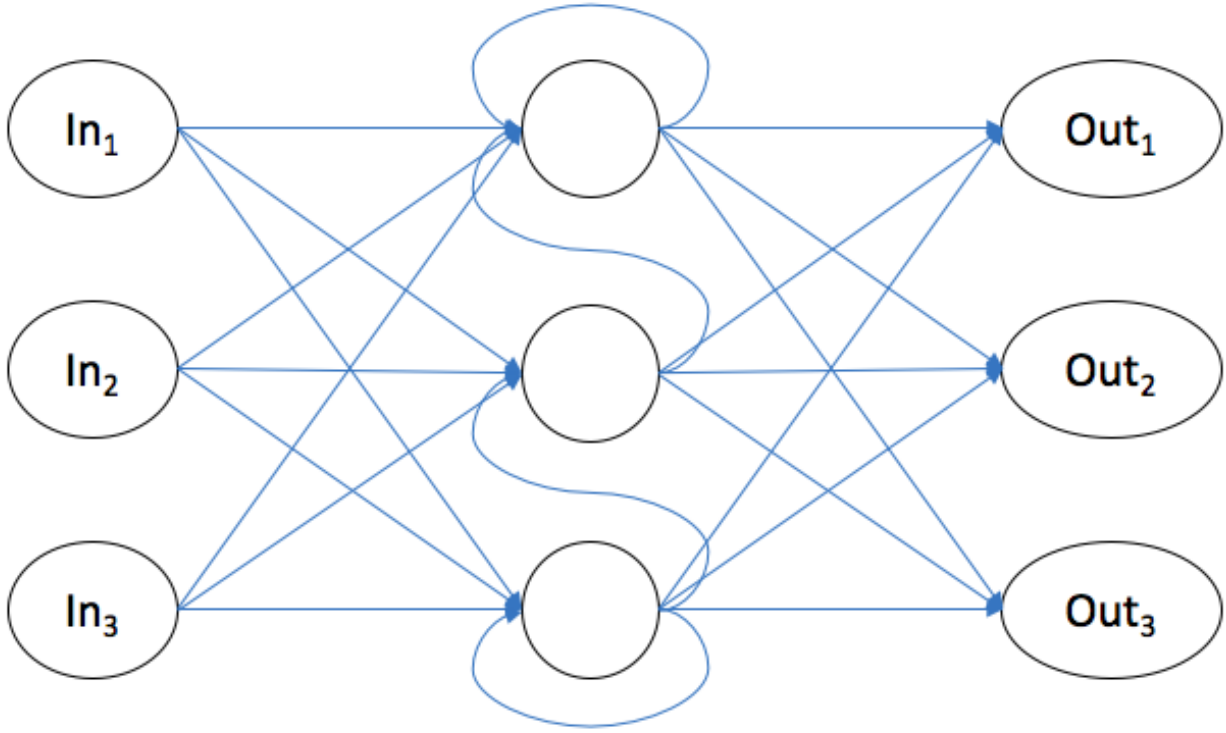


Figure 1: Architecture of Recurrent Neural Network.

Short-Term Memory) allows the users to specify a limit. Backpropagation is the learning algorithm that been used very often in neural networks. Backpropagation first appeared in the work of David et al. [11] 1988. Their work shows that applying backpropagation often results in useful discoveries using gradient descent. Different from Traditional RNN, LSTM only reads the input from the current time when doing a time series prediction which makes it more efficient than the traditional RNN [4]. Figure 2 shows the example of backpropagation structure. Where:

$$Hidden_1 = In_1 \times W_1 + In_2 \times W_4 + In_3 \times W_7$$

$$Out_{Hidden_1} = \frac{1}{1 + e^{-Hidden_1}}$$

$$Out_1 = Out_{Hidden_1} \times W_{10} + Out_{Hidden_2} \times W_{13} + Out_{Hidden_2} \times W_{16}$$

The goal of this step is to find the best weights (W_i) for the neural network to learn.

LSTM is widely used to forecast data in many areas. Kong et al. [7] used LSTM to forecast short-term resident load. Their experiment showed that among all the prediction methods they selected, LSTM has the most accuracy. Huang et al. [6] used the past PM 2.5 concentration and weather report data to predict the PM 2.5 concentration in the future. The result proves the ability of LSTM in predicting PM 2.5. Sagheer et al. [12] developed a model based on LSTM that can deal with most time-series prediction problems. By using the experiment, they verify that their model works well on petroleum time series problems.

3 A NEW SPATIO-TEMPORAL INTERPOLATION METHOD

In this section we propose a novel spatio-temporal interpolation method. This is a general method of spatio-temporal interpolation, but we developed it with the COVID19 pandemic in mind.

3.1 Calculation of Distances between Neighboring States

Next we describe how we calculate the distances between neighboring locations. In the example below we consider the states within the continental USA. First, we find the latitude and the longitude of the centroid of each state as shown in Table 1.

Second, we calculate the distance between two states i and j based on their centroids considering that they lie on the surface of the 3-dimensional earth, as follows. First, let R = 6368 kilometers, and then take:

x_i = R × cos($long_i$) × sin(90°-lat_i)
y_i = R × sin($long_i$) × sin(90°-lat_i)
z_i = R × cos(90°-lat_i)

Similarly, we have:

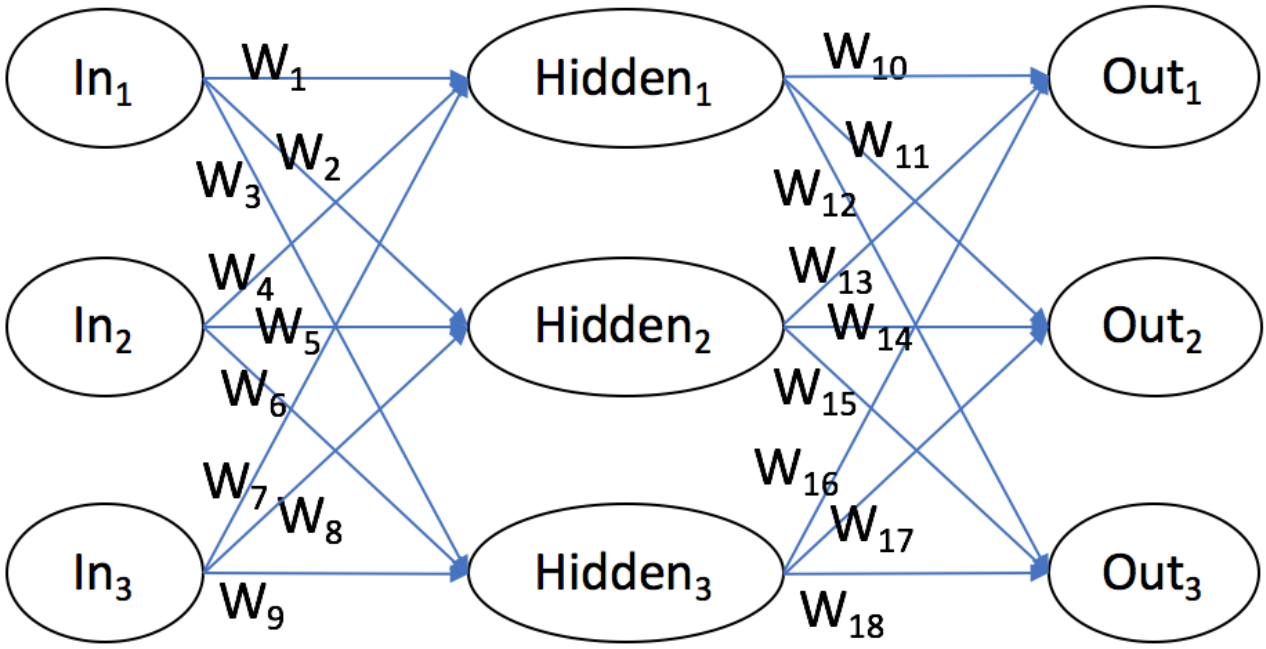


Figure 2: Backpropagation example.

$x_j = \mathrm{R} \times \cos(long_j) \times \sin(90°\text{-}lat_j)$
$y_j = \mathrm{R} \times \sin(long_j) \times \sin(90°\text{-}lat_j)$
$z_j = \mathrm{R} \times \cos(90°\text{-}lat_j)$

Finally, the Euclidean distance in 3-dimensions between the two centroids can be found as follows:

$$distance = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \tag{7}$$

3.2 Proposed Spatio-Temporal Interpolation Method

Intuitively, the number of cases of the COVID-19 pandemic can be better estimated by considering both temporal and spatial interpolations. If a state S has a very high number of COVID-19 cases, then the situation in its neighbors may not affect the development of the number of cases much and could even be ignored because most residents of S will catch the disease from other residents within state S. Therefore, the best temporal interpolation based just on that state's previous cases, denoted as $E_{t,S}$, likely would give the best prediction for the future.

On the other hand, if a state S has few COVID-19 cases relative to its neighbors, then the situation in its neighbors has to be carefully considered because in that case most residents of S could be infected by neighboring state residents when they travel and meet. Therefore, the spatial interpolation based on the neighbor's previous cases, denoted as $E_{s,S}$, likely would give the best prediction for the future.

From the above intuition, it follows that we could estimate the number of cases according to the following formula:

$$E_S = \begin{cases} E_{t,S} & \text{if } E_{t,S} > E_{s,S} \\ E_{s,S} & \text{otherwise} \end{cases} \tag{8}$$

Preliminary experiments suggested that the above still needs to be refined because if one of the neighbors experiences an explosion in the number of cases, then it may not immediately cause an explosion in state S too. In other words, there is some time delay instead of an immediate effect. Therefore, in such cases the temporal interpolation $E_{t,S}$ likely still would give the most accurate prediction while the spatial interpolation $E_{s,S}$ would likely give an overestimate of the number of COVID-19 pandemic cases. Therefore, we need to place some limit on the difference between the two estimates and ignore the spatial estimate if it is excessively larger than the temporal estimate. From the preliminary experiments, we found that the value of 5, which in this case means five cases per one million people, works well as a threshold value. Therefore, we refine the above formula as follows:

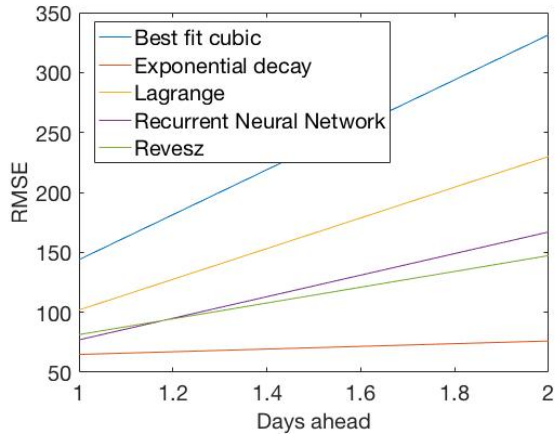
$$E_S = \begin{cases} E_{t,S} & \text{if } E_{t,S} > E_{s,S} \text{ or } E_{s,S} > E_{t,S} + 5 \\ E_{s,S} & \text{otherwise} \end{cases} \tag{9}$$

Table 2: Raw Data of COVID-19 cases taken from the New York Times Github page [14]

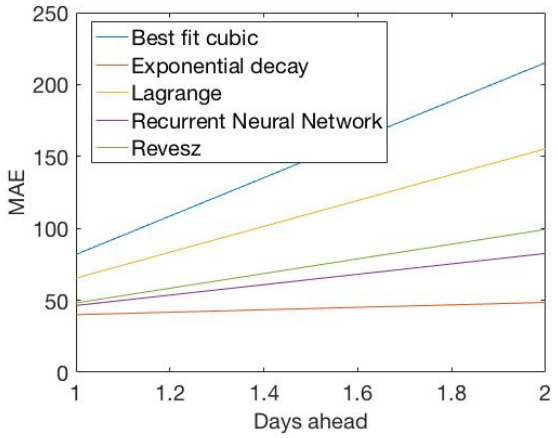
State	5/15	5/16	5/17	5/18	5/19	5/20	5/21	5/22	5/23	5/24
Alabama	272	301	97	315	290	676	362	256	479	329
Arizona	495	462	306	233	396	331	418	293	432	299
Arkansas	97	115	181	54	110	80	455	154	163	147
California	2068	1918	1433	1577	2038	2144	2363	2313	2014	1928
Colorado	394	397	305	265	280	315	392	295	476	208
Connecticut	621	618	716	697	314	587	191	432	382	446
Delaware	150	174	123	199	168	157	192	143	161	119
Florida	928	673	777	854	502	527	1204	776	676	740
Georgia	820	413	299	680	580	948	807	765	674	581
Idaho	37	30	2	34	21	30	28	61	31	24
Illinois	2447	2141	1691	2418	1520	2415	2282	2715	2390	2441
Indiana	654	638	500	478	505	584	687	492	509	477
Iowa	374	279	323	304	341	572	302	340	388	536
Kansas	338	53	7	373	49	173	110	297	51	16
Kentucky	181	195	20	248	164	131	139	213	124	30
Louisiana	348	280	315	334	329	278	1188	421	117	128
Maine	38	45	39	26	28	78	58	71	65	42
Maryland	1084	978	837	962	1782	778	1203	894	1069	1311
Massachusetts	1239	1512	1077	1042	873	1045	1114	805	773	1013
Michigan	493	434	638	799	484	651	480	397	439	312
Minnesota	806	729	699	704	657	641	530	805	840	728
Mississippi	318	322	173	136	272	263	255	402	381	247
Missouri	77	227	275	156	144	141	147	140	125	192
Montana	4	2	0	2	1	7	1	0	0	0
Nebraska	355	454	128	277	221	273	305	238	327	145
Nevada	129	77	125	124	110	132	101	129	238	117
New Hampshire	82	92	40	56	69	147	67	79	75	60
New Jersey	1201	1184	1245	1705	974	1386	1073	1247	385	1050
New Mexico	159	185	91	158	96	146	134	153	170	148
New York	2759	2185	1901	1241	1479	1478	2078	1678	1754	1601
North Carolina	597	819	528	584	641	459	694	746	1070	500
North Dakota	49	87	52	31	63	101	134	88	48	53
Ohio	593	520	449	530	498	484	731	627	613	503
Oklahoma	124	151	73	88	91	43	148	169	111	77
Oregon	62	71	11	64	39	75	16	47	24	39
Pennsylvania	1020	1050	517	931	728	747	1101	1053	797	554
Rhode Island	203	215	240	121	156	405	215	165	216	113
South Carolina	218	254	155	126	114	119	204	259	257	201
South Dakota	95	72	28	40	58	92	0	179	112	95
Tennessee	329	287	187	376	534	233	436	394	401	351
Texas	1818	879	962	960	1292	1103	1258	928	1017	664
Utah	170	138	170	146	146	187	159	181	203	132
Vermont	1	1	6	0	4	0	6	2	2	2
Virginia	859	1011	705	752	1005	763	1229	813	799	495
Washington	264	196	172	240	225	118	169	241	213	245
West Virginia	13	23	22	10	12	53	36	102	24	42
Wisconsin	574	514	203	151	279	573	318	665	428	377
Wyoming	15	25	13	12	10	11	14	2	10	0

Table 3: Moving Average of Five Days Data of COVID-19 cases (The dates represent the middle of five days.)

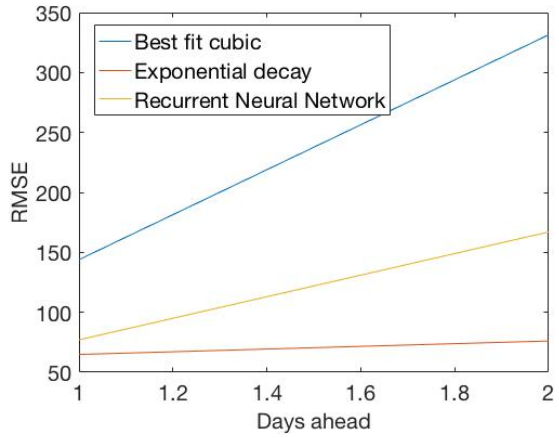
State	5/15	5/16	5/17	5/18	5/19	5/20	5/21	5/22
Alabama	261.4	277.2	255	335.8	348	379.8	412.6	420.4
Arizona	440.2	398.8	378.4	345.6	336.8	334.2	374	354.6
Arkansas	119	115.4	111.4	108	176	170.6	192.4	199.8
California	1843.2	1745	1806.8	1822	1911	2087	2174.4	2152.4
Colorado	361.2	346.4	328.2	312.4	311.4	309.4	351.6	337.2
Connecticut	617.2	652.2	593.2	586.4	501	444.2	381.2	407.6
Delaware	185.8	183.4	162.8	164.2	167.8	171.8	164.2	154.4
Florida	733	808	746.8	666.6	772.8	772.6	737	784.6
Georgia	528.6	553.6	558.4	584	662.8	756	754.8	755
Idaho	25.4	26	24.8	23.4	23	34.8	34.2	34.8
Illinois	2238.4	2380.8	2043.4	2037	2065.2	2270	2264.4	2448.6
Indiana	548.6	570.6	555	541	550.8	549.2	555.4	549.8
Iowa	347.8	333.2	324.2	363.8	368.4	371.8	388.6	427.6
Kansas	142.8	177.4	164	131	142.4	200.4	136	129.4
Kentucky	158	187.8	161.6	151.6	140.4	179	154.2	127.4
Louisiana	476.4	420.8	321.2	307.2	488.8	510	466.6	426.4
Maine	42	39.6	35.2	43.2	45.8	52.2	60	62.8
Maryland	949.2	991	1128.6	1067.4	1112.4	1123.8	1145.2	1051
Massachusetts	1335.6	1311	1148.6	1109.8	1030.2	975.8	922	950
Michigan	621.6	709.2	569.6	601.2	610.4	562.2	490.2	455.8
Minnesota	636.8	693	719	686	646.2	667.4	694.6	708.8
Mississippi	277.6	268.4	244.2	233.2	219.8	265.6	314.6	309.6
Missouri	189	171.2	175.8	188.6	172.6	145.6	139.4	149
Montana	1.4	1.6	1.8	2.4	2.2	2.2	1.8	1.6
Nebraska	323.2	310.6	287	270.6	240.8	262.8	272.8	257.6
Nevada	124.8	115.4	113	113.6	118.4	119.2	142	143.4
New Hampshire	71.4	70.6	67.8	80.8	75.8	83.6	87.4	85.6
New Jersey	1118.2	1295.8	1261.8	1298.8	1276.6	1277	1013	1028.2
New Mexico	145.2	146.4	137.8	135.2	125	137.4	139.8	150.2
New York	2266.4	2090	1913	1656.8	1635.4	1590.8	1693.4	1717.8
North Carolina	627	654.2	633.8	606.2	581.2	624.8	722	693.8
North Dakota	65.8	56.8	56.4	66.8	76.2	83.4	86.8	84.8
Ohio	533.6	545.6	518	496.2	538.4	574	590.6	591.6
Oklahoma	115.8	109.2	105.4	89.2	88.6	107.8	112.4	109.6
Oregon	52.8	54.2	49.4	52	41	48.2	40.2	40.2
Pennsylvania	867.6	896.4	849.2	794.6	804.8	912	885.2	850.4
Rhode Island	212	192	187	227.4	227.4	212.4	231.4	222.8
South Carolina	177.8	182.4	173.4	153.6	143.6	164.4	190.6	208
South Dakota	64.8	59	58.6	58	43.6	73.8	88.2	95.6
Tennessee	322.8	301.2	342.6	323.4	353.2	394.6	399.6	363
Texas	1296	1193.2	1182.2	1039.2	1115	1108.2	1119.6	994
Utah	156.8	152	154	157.4	161.6	163.8	175.2	172.4
Vermont	2.6	2.2	2.4	2.2	3.2	2.4	2.8	2.4
Virginia	917.6	878.8	866.4	847.2	890.8	912.4	921.8	819.8
Washington	219	246.8	219.4	190.2	184.8	198.6	193.2	197.2
West Virginia	22.8	19.6	16	24	26.6	42.6	45.4	51.4
Wisconsin	390.8	363.8	344.2	344	304.8	397.2	452.6	472.2
Wyoming	15.8	15.6	15	14.2	12	9.8	9.4	7.4



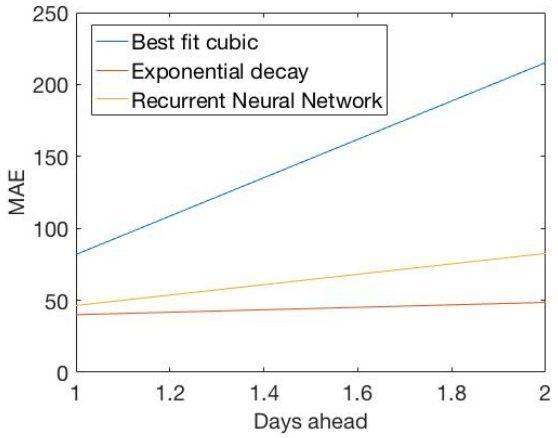
(a) Overall RMSE



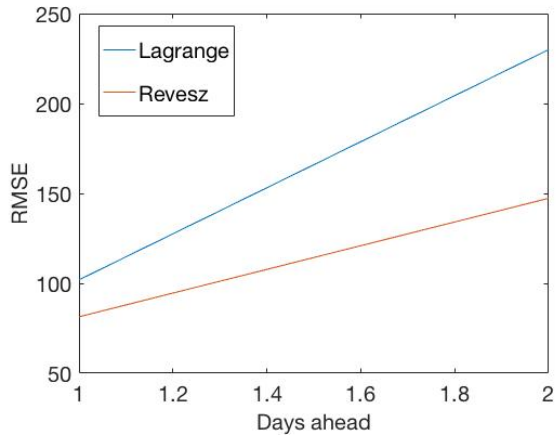
(b) Overall MAE



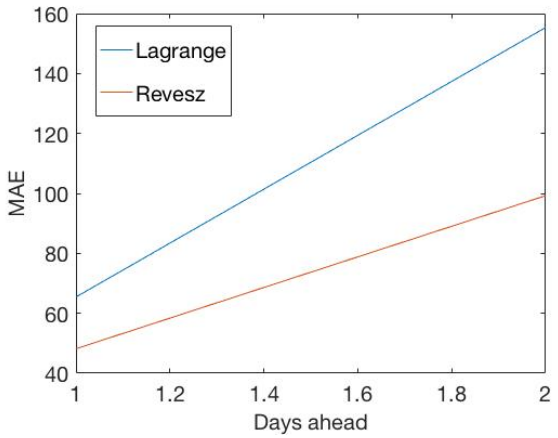
(c) RMSE 6 inputs group



(d) MAE 6 inputs group



(e) RMSE 3 inputs group



(f) MAE 3 inputs group

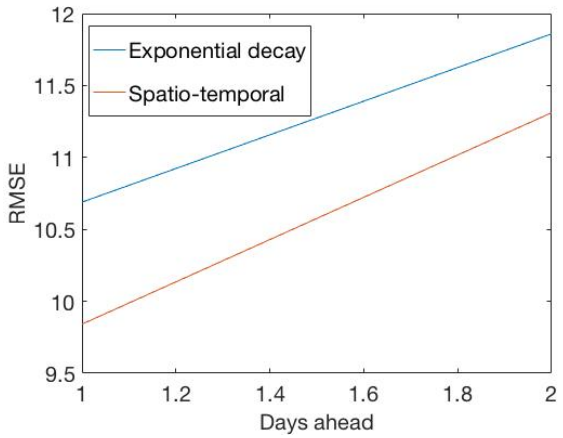
Figure 3: Comparison of the RMSEs and MAEs of the temporal interpo-lation methods and RNN.

Table 4: RMSEs and MAEs of the temporal interpolation methods and RNN

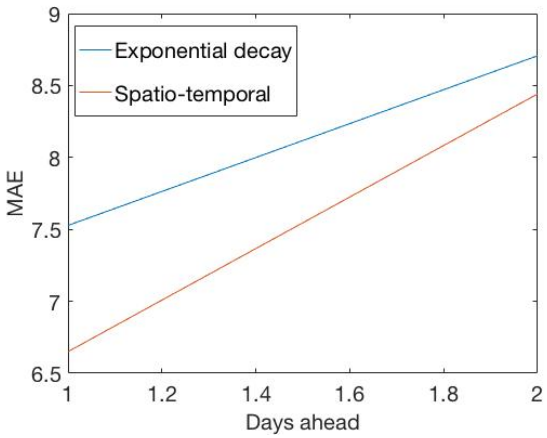
	RMSE		MAE	
	1 day ahead	2 days ahead	1 day ahead	2 days ahead
Best Fit Cubic	143.9267963	331.4393394	81.87361111	215.2199074
Exponential decay	64.8459129	76.08476497	39.99752604	48.55071615
Lagrange	101.9343661	229.9731089	65.44166667	155.3458333
Recurrent Neural Network	77.02574416	166.9927021	46.40645103	82.57963644
Revesz	81.40245185	147.2905943	48.17916667	99.25416667

Table 5: RMSEs and MAEs of the exponential decay and the new spatio-temporal interpolation methods

	RMSE		MAE	
	1 day ahead	2 days ahead	1 day ahead	2 days ahead
Exponential decay	10.68921654	11.85923055	7.527681653	8.707041576
Spatio-temporal	9.841357179	11.31102594	6.650146028	8.44158513



(a) Comparison of the RMSEs of the exponential decay and the new spatio-temporal interpolation methods.



(b) Comparison of the MAEs of the exponential decay and the new spatio-temporal interpolation methods

Figure 4: Comparison of the RMSEs and MAEs of the exponential decay and the new spatio-temporal interpolation methods.

4 EXPERIMENTAL RESULTS

4.1 Data Collection and Processing

We will use the number of COVID-19 cases in each state to test how well the new method predicts the number of cases of the COVID-19 pandemic of future days. From [14] we collected data of new cases every day for all the states in the US except Alaska and Hawaii as shown in Table 2.

Table 3 shows the data after the moving average, the data shown in the first row is the middle day of the moving average.

4.2 Evaluating the Temporal Interpolation Methods

For the experiment, we will divide the methods into two groups: for the Lagrange method and the method introduced by Revesz, we use three inputs, for Exponential Decay, Best fit cubic and Recurrent Neural Network, we use six inputs.

Table 4 shows the RMSEs and MAEs of temporal interpolation methods and RNN for predicting the moving average of the COVID-19 pandemic cases.

Figure 3a shows the RMSEs of the temporal interpolation methods for predicting the moving average of the COVID-19 pandemic cases one and two days ahead. In this case all RMSEs were below 350. The best temporal method was the exponential decay interpolation method.

Figure 3b shows the MAEs of the temporal interpolation methods for predicting the moving average of the COVID-19 pandemic cases one and two days ahead. Similarly to the previous figure, this also shows that the exponential decay interpolation method was the most accurate temporal interpolation method.

Figure 3c and 3d show the RMSEs and MAEs of the temporal interpolation methods for predicting the moving average of the COVID-19 pandemic cases one and two days ahead with a six inputs group. Figure 3e and 3f show the RMSEs and MAEs of a three inputs group. Ideally, the methods with more inputs should have higher accuracy than the methods with fewer inputs. However, the result shows that overall, the methods with three inputs have more accurate results than the methods with six inputs.

4.3 Evaluating the Novel Spatio-Temporal Interpolation Method

In this section we use the number of cases per one million people instead of the total number of cases. This change in units does affect the relative performance of the temporal interpolation methods, meaning that the exponential decay interpolation method is still the most accurate among the tested temporal interpolation methods. Figures 4 compares the exponential decay temporal interpolation method with our novel spatio-temporal interpolation method described in Section 3.2. Figure 4a shows that the exponential decay interpolation method had a RMSE of 11.86 for two days ahead, while the new spatio-temporal interpolation method had a RMSE of 11.31. Moreover, the exponential decay had a RMSE of 10.69 for one day ahead versus the new spatio-temporal interpolation had a RMSE of only 9.84.

Table 5 shows the RMSEs and MAEs of Exponential decay and Novel Spatio-Temporal Interpolation Method for predicting the moving average of the COVID-19 pandemic cases.

Figure 4b shows that the exponential decay interpolation method had a MAE of 8.71 for two days ahead, while the new spatio-temporal interpolation method had a MAE of 8.44. Moreover, the exponential decay had a MAE of 7.53 for one day ahead versus the new spatio-temporal interpolation had a MAE of only 6.65.

These experimental results clearly show that the new spatio-temporal interpolation method is better than any of the pure temporal interpolation methods. Moreover, the spatio-temporal interpolation method is also more accurate than just the IDW spatial interpolation method. In fact, the IDW spatial interpolation method had a RMSE of 39.2 and a MAE of 29.78 for one day ahead. for one day ahead

4.4 Discussion of the Experimental Results

A surprising result was that the RNN did not perform better than all the temporal methods. The reason seems to be that RNNs generally need more input data than the other temporal methods. Hence in a fair comparison, when all the methods are given the same data, the RNN has a relative disadvantage compared to the other methods. There are many cases when users of these methods do not have much data available. This is the case for COVID-19 because it is a new pandemic.

The new spatio-temporal method performed better than the exponential decay method because the spread of the COVID-19 virus is a spatial phenomenon. When large numbers of people who already have COVID-19 travel from one state to another, then the primary cause of the spread of the pandemic could be the outside visitors instead of the local people spreading the disease to their neighbors. If the effect of the visitors is the primary cause of the spread, then the spatio-temporal method adjusts its estimate accordingly while the exponential decay method does not do that. This lack of adjustment is a cause for the great inaccuracy in the exponential decay prediction of the number of cases on some days when the pandemic is in its early stage. Unfortunately, predicting the early stages accurately is the most crucial for epidemiologists because they need to allocate resources and prepare the population for the spread of the disease in the early stage.

The experimental results demonstrated the benefit of combining some temporal interpolation methods with some spatial interpolation methods in order to achieve a higher prediction accuracy. Naturally, we chose the best performing temporal interpolation method. However, the spatio-temporal method could be further improved if we used some other spatial interpolation method instead of the IDW method. We chose the IDW method only because it is widely popular and easy to implement. As a future work, it would be good to experiment with several alternative spatial interpolation methods, and then chose the best of those methods.

5 CONCLUSIONS AND FUTURE WORK

We proposed a new spatio-temporal prediction method for the number of cases of the COVID-19 pandemic. The need for a new prediction method arose for two reasons. First, the temporal and the spatial interpolation methods that we tested did not give very good results. Second, we detected an inherent spatio-temporal nature in this problem but with the caveat that sometimes the problem seems more temporal and essentially a local phenomenon, while sometimes it seems more spatial with a heavy interaction among neighbors. In other words, there is something in the nature of the COVID-19 pandemic that is different from other spatio-temporal interpolation problems such as the prediction of house prices over time within a city [8] or the spread of other epidemics such as the West Nile Virus [10].

Our experimental results show that the new spatio-temporal method is a sensible combination of the best temporal interpolation method and the IDW interpolation method. It is likely that if we tested more spatial interpolation methods, or we tested more p values for the IDW, then we would have been able to find a better spatial interpolation algorithm than the current IDW with $p = 1$. Nevertheless, the main idea of the combination as outlined in Section 3.1 has proven to be a good idea that can just get even better results as the temporal and the spatial components also improve in accuracy. Hence to make further improvements, we plan to test both more temporal and spatial interpolation methods in the future. There is especially a need to test several spatial interpolation methods because, as we mentioned earlier, we simply took the IDW method because it seems to be a popular method. However, we need to make sure that it has the best performance among comparable spatial interpolation methods.

Finally, although the COVID-19 pandemic is a current topic that was the main focus of our experiments, in the future we also plan to apply the new spatio-temporal method for other types of data. For example, we are interested in testing whether the method is also good at predicting the outcome of elections [2, 3].

REFERENCES

[1] Richard L Burden, J Douglas Faires, and Albert C Reynolds. 2001. Numerical analysis.

[2] J Gao and P. Z. Revesz. 2005. Adaptive spatio-temporal interpolation methods. In *Proc. of the 1st International Conference on Geometric Modeling, Visualization & Graphics*. 1622–1625.

[3] J. Gao and P. Z. Revesz. 2006. Voting prediction using new spatiotemporal interpolation methods. In *Proceedings of the 7th Annual International Conference on Digital Government Research, DG.O 2006, San Diego, California, USA, May 21-24, 2006 (ACM International Conference Proceeding Series)*, José A. B. Fortes and Ann Macintosh (Eds.), Vol. 151. Digital Government Research Center, 293–300. https://doi.org/10.1145/1146598.1146678

[4] Felix Gers. 2001. *Long short-term memory in recurrent neural networks*. Ph.D. Dissertation. Verlag nicht ermittelbar.

[5] Chaolin Huang, Yeming Wang, Xingwang Li, Lili Ren, Jianping Zhao, Yi Hu, Li Zhang, Guohui Fan, Jiuyang Xu, Xiaoying Gu, et al. 2020. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *The lancet* 395, 10223 (2020), 497–506.

[6] Chiou-Jye Huang and Ping-Huan Kuo. 2018. A deep cnn-lstm model for particulate matter (PM2. 5) forecasting in smart cities. *Sensors* 18, 7 (2018), 2220.

[7] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang. 2019. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid* 10, 1 (2019), 841–851.

[8] L. Li and P. Z. Revesz. 2004. Interpolation methods for spatio-temporal geographic data. *Comput. Environ. Urban Syst.* 28, 3 (2004), 201–227. https://doi.org/10.1016/S0198-9715(03)00018-8

[9] Peter Z Revesz. 2017. Data Mining Citations to Predict Emerging Scientific Leaders and Citation Curves. *International Journal of Education and Information Technologies* 11 (2017), 171–179.

[10] P. Z. Revesz and S. Wu. 2006. Spatiotemporal reasoning about epidemiological data. *Artif. Intell. Medicine* 38, 2 (2006), 157–170. https://doi.org/10.1016/j.artmed.2006.05.001

[11] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1988. *Learning Representations by Back-Propagating Errors*. MIT Press, Cambridge, MA, USA, 696–699.

[12] Alaa Sagheer and Mostafa Kotb. 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing* 323 (2019), 203–213.

[13] Donald Shepard. 1968. A Two-Dimensional Interpolation Function for Irregularly-Spaced Data. In *Proceedings of the 1968 23rd ACM National Conference (ACM '68)*. Association for Computing Machinery, New York, NY, USA, 517–524. https://doi.org/10.1145/800186.810616

[14] M. Smith, K. Yourish, and S. et al. Almukhtar. 2020. *Coronavirus (COVID-19) Data in the United States, The New York Times*. https://github.com/nytimes/covid-19-data

A Pattern-based Approach for an Early Detection of Popular Twitter Accounts

Jonathan Debure
AIRBUS & CNAM, Paris, France
jonathan.debure@airbus.com

Stephan Brunessaux
AIRBUS, Paris, France
stephan.brunessaux@airbus.com

Camelia Constantin
Sorbonne University, Paris, France
camelia.constantin@lip6.fr

Cédric du Mouza
CNAM, Paris, France
dumouza@cnam.fr

ABSTRACT

Social networks (SN) are omnipresent in our lives today. Not all users have the same behaviour on these networks. If some have a low activity, rarely posting messages and following few users, some others at the other extreme have a significant activity, with many followers and regularly posts. The important role of these popular SN users makes them the target of many applications for example for content monitoring or advertising. It is therefore relevant to be able to predict as soon as possible which SN users will become popular.

In this work, we propose a technique for early detection of such users based on the identification of characteristic patterns. We present an index, H^2M, which allows a scaling up of our approach to large social networks. We also describe our first experiments that confirm the validity of our approach.

CCS CONCEPTS

• **Information systems → Social networks**.

KEYWORDS

Twitter, popularity detection, pattern matching

ACM Reference Format:
Jonathan Debure, Stephan Brunessaux, Camelia Constantin, and Cédric du Mouza. 2020. A Pattern-based Approach for an Early Detection of Popular Twitter Accounts. In *24th International Database Engineering Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3410566.3410600

1 INTRODUCTION

Online social networks have become nowadays an essential means for communication, entertainment and marketing. Platforms like YouTube, Facebook, Twitter and Instagram gather hundreds of millions of users every day. While they have their own specifics and propose different content and interactions ways, these platforms share some common characteristics: first, their large number of users and the phenomenal amount of data (texts, pictures, videos, etc) produced daily; second, their network structure, with users connected to other users to share content; third, their high dynamicity with new users joining the platforms, others leaving, and connections between users which are continuously created or deleted.

These different characteristics make these platforms a tool particularly used to communicate information to a large number of people. In these networks, the most popular and influential users have quickly been the center of attention for many applications, since they will accelerate the spread of information to the greatest number of users [8]. For instance, for online advertising campaigns on social networks or on the Web, advertisers seek to place their advertisements among the users who have the most visibility in order to reach a maximum of people [2, 5, 9]. Likewise, for marketing purposes, highly followed users, called *influencers*, are paid to test and promote different products. In another area, popular users allow messages to be transmitted to a large audience for which social networks are the main media of information. These are the users who can quickly spread fake news or on the contrary bring a denial [7, 31]. Checking the content they publish is therefore particularly important. In the area of security, monitoring the content posted by some popular users who use social media for propaganda and / or indoctrination is also essential.

The various existing works offer techniques for detecting users who are already popular or influential in social networks. However, the various examples of applications presented above show that it is important to be able to identify the appearance of popular users on social networks as soon as possible. This article is, to our knowledge, the first to try to identify users who are on the way to more or less near future, to become popular. By detecting recurring patterns in the evolution of the popularity of accounts becoming popular, we manage with good precision to detect users several weeks before they become really popular. In addition, the index structure that we offer makes it possible to scale up to hundreds of millions of users and therefore allow our solution to be deployed for real social media platforms. Our experiences with real Twitter datasets validate our approach.

In summary, the contributions of our article are as follows:

(1) a characterization of the evolution of popularity for different classes of users (popular, non-popular, becoming popular);
(2) a pattern-based approach for early detection of popular users;
(3) an indexing structure for an efficient pattern-matching which scales to hundreds of millions of users to an early detection of future popular users;
(4) a validation on a large real Twitter dataset.

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 describes the data model for popularity evolution. An analysis of a real Twitter dataset and the patterns we extracted for different classes of users is presented in Section 4. We introduce our pattern-based approach along with its indexing structure for an early detection of users becoming popular in Section 5. Section 6 gathers some of the experiments we perform to validate our approach. We conclude the paper and introduce some future work in Section 7.

2 RELATED WORK

Most of the recent work trying to estimate the popularity of accounts in social networks have focused on influence spreading ([18], [21], [20], [19]) as a measure for popularity. Basically these works observe how many users along with their distance (number of intermediate users between the user who produces the news and the one who receives them) will receive an item produced by a user. They rely consequently on the topology and on the eventuality (based on his homophily, his topics of interests, his activity, etc) for a user to propagate this item. Some works try not only to study the propagation but also to capture and to estimate the ability of some users to trigger actions or change an opinion in the network, generally their direct neighborhood. Different influence studies propose to score users ([1],[27],[28]) to rank them and to highlight principal node of a social network.

Users popularity analysis in social media, especially in social networks, has taken importance in the last decade because they appear to be a rather precise way to estimate the opinion of a pool of users. For instance, during American elections, online social media have been used by politician to spread their program [17]. Different analyses of the social networks try to determine the most popular candidate and to predict the elections results [10], [15]. Similarly, social media analyses try to predict future popular content [26], like hot news [6] and try to understand hidden mechanisms. Multiple studies try to find how information are spread on Twitter, with probabilistic model [30] and others study the retweet action from Twitter [14] as an indicator of users popularity. In [24], the authors explain that retweet is time-sensitive. The main difficulty for these approaches is that they assume we have a knowledge of the content published or liked by any user, and also enough information to estimate their opinion or interests which is most of the time either very costly or impossible to get in real social network systems, except for the platform owners.

A solution for popular user detection and prediction, that we adopt in this work, is to identify features which characterize popular users, based on a large sample of users, and to perform machine learning techniques to identify other popular users on the global dataset and/or to perform predictions to early detect their apparition. We propose in this paper to extract patterns which characterize a class of popularity for users. Patterns mining is a popular method to find frequent occurring patterns. Patterns mining is divided in two majors models: item-set mining and string-mining. Item-set mining focuses on detecting frequent item-set and is mostly used in database mining. The two most popular algorithms for item-set mining are APRIORI [25] and FPGrowth [13]. APRIORI extracts all item-sets of a specific length respecting a minimum support, and then scans transactions at each iteration that makes this algorithm rather slow. FPGrowth is an improved APRIORI algorithm which relies on a Tree (FP Tree) to mine frequent item-set and which only needs to scan database twice. Item-set mining is also used for association rules mining [4], *i.e.* the extraction of correlation between items. String-mining is a distinct pattern mining technique, which consists in detecting frequents patterns in alphanumerical sequence generally very long. This technique is mainly use in bioinformatics[3] to analyze DNA/RNA and to detect or to extract proteins from nucleotide sequences. String mining is mostly represented by three algorithms: SPADE [29] which is a vertical sequential patterns mining, FreeSpan [11] which partitions the search space and projects sequences and PrefixSpan [12] that is based on FreeSpan but avoids to check every possible combination.

3 THE DATA MODEL

We introduce in this section our notations and our data model. We consider the Twitter platform and its underlying directed graph $(\mathcal{U}, \mathcal{F})$ where $\mathcal{U}$ denotes the set of nodes, *i.e.* users, and $\mathcal{F} \subseteq \mathcal{U}\times\mathcal{U}$ is the set of edge, such as $(u_1, u_2) \in \mathcal{F}$ means user u_2 follows user u_1.

3.1 Account popularity

Remind that our objective is to perform early detection of (future) popular users. Assuming the existence of the function $Follow : \mathcal{U} \times [0, T] \rightarrow \mathbb{N}$ which returns the number of followers for an account $u \in \mathcal{U}$ at the instant $t \in [0, T]$, we adopt in this paper the following definition of popularity:

Definition 1. *[Popularity] The popularity of an account corresponds to its visibility, that means how many persons can read, comment, propagate a message that this account produces. It is here simply estimated by the number of followers $Follow(u, t)$ an account u possesses at the instant t.*

With respect to this definition we propose the following account classification:

- Non-popular accounts: this class regroups users that are never popular all along the observation period $[0, T]$. So, assuming an unpopularity threshold φ, these accounts verify that: $u \in \mathcal{U}, \forall t \in [0, T] : Follow(u, t) \leq \varphi$
- Popular accounts: this class corresponds to users that are already popular on our study period $[0, T]$. So assuming a popularity threshold ε, this implies that at $t = 0$ we have $Follow(u, 0) \geq \varepsilon$.
- Becoming popular accounts: this class regroups users that are not popular at the beginning of our period $[0, T]$ but are popular at the end. So, based on our two thresholds ε and φ, it corresponds to users $u \in \mathcal{U}$ such as $Follow(u, 0) \leq \varphi$ and $Follow(u, T) \geq \varepsilon$.

3.2 Popularity evolution

We assume that the platform periodically updates its statistics for each user. The period between two updates is constant and is considered in the following as our indivisible time unit. So at each

time instant $t \in [0, T]$, we report for each account u the number of followers. To estimate the popularity evolution, we compute the gain in number of followers between two time instants. To reduce the impact of the size of the accounts, we propose to use the log function on the gain since a gain of 10k followers should be considered as a gain of the same order as a gain of 20k followers when the user has a popular account. Due to the domain definition of the log function we use the following definition for the *gain* function.

Definition 2. *[Popularity gain] Consider an instant $t \in [0, T-1]$, the popularity gain for a user $u \in \mathcal{U}$ is:*

$$gain(u,t) = \begin{cases} \log(Follow(u,t) - Follow(u,t-1)) \\ \qquad if Follow(u,t) > Follow(u,t-1) + 1 \\ 0 \quad if |Follow(u,t) - Follow(u,t-1)| \leq 1 \\ -\log(Follow(u,t-1) - Follow(u,t)) \\ \qquad if Follow(u,t-1) > Follow(u,t) + 1 \end{cases}$$

Consequently, the popularity evolution for an account over a given period $[0, T]$ corresponds to a time series made up of the popularity gain for each time unit. We formally adapt the following definition for the popularity evolution.

Definition 3. *[Popularity evolution] The popularity evolution of user $u \in \mathcal{U}$ on the time period $[0, T]$ is represented by the time series*

$$\pi(u) =< gain(u,1), gain(u,2), \ldots, gain(u,T) >$$

.

We denote with Π the set of the popularity evolution for all users from $\mathcal{U}$.

This raw time series is interesting to extract some statistics about the given dataset but is not suitable to compare users in order to cluster them and to detect classes of users with some specific behaviors, or oppositely to identify users who have a divergent behavior. In order to achieve these goals, a traditional approach (see for instance the SAX approach [16, 22]) consists in encoding the time series using a limited set of symbols.

So we assume the existence of an alphabet of symbols Ω for the encoding and of a mapping function $mapping : \mathbb{R} \rightarrow \Omega$. Defining the size of the alphabet and the mapping function is a difficult task which must highly rely on the properties of the studied dataset. The size of the alphabet used for the encoding sets a level of refinement for the encoded sequence. Indeed, a large alphabet will provide more precision about the popularity evolution, but it reduces the number of similarities between sequences detected. Oppositely, a small alphabet will lead to the extraction of numerous similar subsequences between the different sequences, while they correspond in reality to an evolution relatively different. Similarly the mapping function will highly impact the similarity detection between sequences. When too many different gain values are mapped to the same symbols, while other symbols correspond to very few gains, it results in an issue similar to the use of a small alphabet: similar subsequences that are detected may correspond to very different behaviors. So the choice of the alphabet and the mapping function has an important impact on the results. We do not investigate further this problem, but we take it into consideration when proposing our alphabet and mapping function in our analysis and experimental sections.

Based on this representation, we want to study whether some subsequences, we will call *patterns* in the following, are characteristic of a popularity class. If such subsequences exist, we expect them to allow us to perform early detection of emerging popular accounts.

Assume a *contains* function, $contains : 2^{\Omega} \times \mathcal{S} \rightarrow \mathbb{N}$, with 2^{Ω} denoting the powerset of Ω, where $contains(x, y)$ returns 1 the sequence y contains the word x, and 0 otherwise. Then we adopt the following definition for a popularity pattern.

Definition 4. *[Popularity pattern] Consider a given size value σ and assume the existence of a relevance threshold Γ, a popularity pattern of size σ is a word $p \in \Omega^{\sigma}$ such that*

$$\frac{\sum_{s \in \mathcal{S}} contains(p, s)}{|\mathcal{S}|} \geq \Gamma$$

The relevance threshold Γ allows to set a minimal support for a pattern. In other words, we only keep patterns which are significant because they are enough present in several sequences.

We denote $\mathcal{S}_{\Pi}$ the set of all the popularity patterns for a popularity evolution set Π. The sets of *exclusive* popularity patterns for the restrictions to the popular, unpopular and becoming popular evolution sets are denoted $\mathcal{S}_{\Pi^+}$, $\mathcal{S}_{\Pi^-}$ and $\mathcal{S}_{\Pi^*}$ respectively. By *exclusive* we mean the popularity patterns which are present only in the considered class of users.

4 POPULARITY ANALYSIS

This section aimed at analyzing a real Twitter dataset to check the existence of such characteristic evolution patterns in our three classes of users. We first introduce our dataset along with its main features, then we extract the popularity patterns for each class and we analyze these different sets of patterns.

4.1 Presentation of our dataset

To build our dataset we use the Twitter API Stream that allows us to collect 1% of all tweets published on the platform. For our dataset, we only collect users metadata and we select users that had a sufficient activity, *i.e.* at least 3 tweets, during our observation period of 36 weeks. We obtain a dataset consisting of around 32.9M users along with 150M tweets.

Then we set the two thresholds φ and ε necessary to determine our three groups of users according to their popularity. Based on the follower distribution of our dataset depicted in Figure 1, we decide arbitrarily that an account with less than $\varphi = 400$ followers is considered as unpopular, while an account with more than $\varepsilon = 2,000$ is identified as a popular account. Thus, the class of *becoming popular* accounts corresponds to the accounts which have less than 400 followers at the beginning of our observation period and more than 2,000 at the end. We report in Table 1 the size of each class of accounts in our dataset that we will investigate in the following. As expected, most accounts belong to the non-popular class and present a low activity with 3-4 tweets generally on the period of observation. The 2.1 million popular accounts have a more important activity, but there is an important discrepency between accounts as it is enlightened by a high standard deviation. In fact, this class is quite heterogeneous with for instance news agency

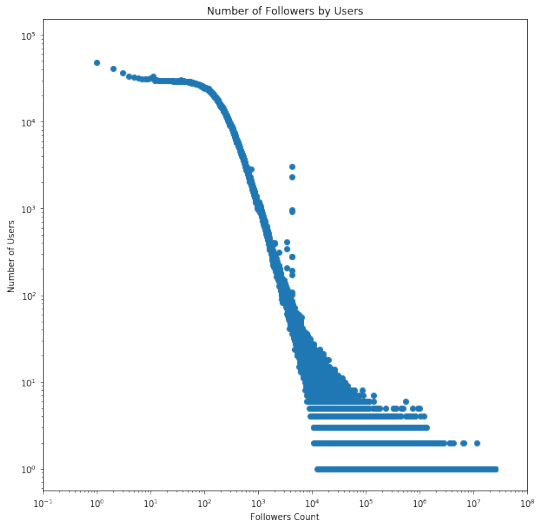


Figure 1: Follower distribution

Table 1: Sub-datasets for each class of users

	global	non-popular	becoming pop.	popular
# accounts	3.2289e+07	3.0106e+07	7.7364e+04	2.1056e+06
# tweets	1.4984e+08	9.2258e+07	3.6330e+06	5.3951e+07
tweets: mean	4.64	3.06	46.96	25.62

or brand accounts with an important activity, and popular personalities (actors, singers, etc) with many followers but few tweets. Accounts that are not popular at the beginning of the observation period but then become popular are not as numerous, but they have a higher activity. This can somehow explain that they become popular on a specific topic because they publish more on this topic what attracts followers.

Table 2 shows the number of followers for each class of users. In our dataset the average number of followers is 1,428, which is more than the last published estimation [1] due to our 3-week threshold for the user activity which discards accounts with few followers and extremely low activity (especially remember that we use the Twitter API with only 1% of the messages). We observe that non-popular accounts have a quite stable number of followers. Oppositely, we see that while becoming popular accounts and popular have generally between 2.10^3 and 3.10^4 followers, the last decile of users reaches respectively millions and dozens of millions of followers, with a more important discrepancy for popular accounts.

4.2 Pattern extraction

In order to perform our pattern extraction, we have first to set the size of the alphabet Ω and to determine the *mapping* function used for the popularity evolution encoding. As explained above, the *mapping* function should, as much as possible, uniformly distribute the gain values on the different symbols of Ω.

We compare different sizes of alphabet for Ω and we finally choose $|\Omega| = 8$. Remember that a small alphabet do not allow

[1] https://www.brandwatch.com/blog/twitter-stats-and-statistics/

Table 2: Followers for each class of users

	global	non-popular	becoming pop. at t=0	becoming pop. at t=T	popular
mean	1.4280e+03	1.2737e+02	1.4059e+02	8.2574e+03	2.2431e+04
std	5.2869e+04	1.1283e+02	1.2261e+02	4.9651e+04	2.3761e+05
min	2.4000e+01	0.0000e+00	0.0000e+00	2.0010e+03	2.0010e+03
25%	5.2000e+01	2.8000e+01	2.7000e+01	2.4420e+03	2.9330e+03
50%	1.8800e+02	9.6000e+01	1.1050e+02	3.3160e+03	4.7430e+03
75%	4.8800e+02	2.0800e+02	2.4000e+02	5.9640e+03	1.0699e+04
85%	8.1900e+02	2.7000e+02	3.0100e+02	9.4710e+03	1.8553e+04
90%	1.2040e+03	3.0700e+02	3.3300e+02	1.2800e+04	2.8601e+04
max	7.7129e+07	3.9900e+02	3.9900e+02	6.7735e+06	7.7129e+07

symbol	A	B	C	D	E	F	G	H
range	$]-\infty,-2]$	$[-2,-0.7]$	$[-0.7,0.7]$	$[0.7,1.6]$	$[1.6,2]$	$[2,2.7]$	$[2.7,3]$	$[3,\infty[$

Table 3: Popularity evolution encoding

Table 4: Symbols distribution

symbol	Becoming Pop.	Popular	Non-popular	Global
A	3.94%	3.39%	0.0%	1.39%
B	7.20%	16.50%	0.60%	4.94%
C	6.74%	20.27%	86.7%	59.14%
D	14.72%	29.88%	12.30%	16.05%
E	18.36%	12.02%	0.33%	5.91%
F	36.55%	12.08%	0.05%	9.15%
G	6.64%	2.58%	0.0%	1.72%
H	5.82%	3.24%	0.0%	1.69%

to capture significant patterns since they will cover very distinct behaviors, while a large alphabet provides very precise patterns but these patterns only correspond to a very small number of sequences.

We report in Table 3 our implementation of the *mapping* function for the computed gain values according to Definition 2.

Then we compute the gain of followers for each week of the 36-week observation period and we apply the *mapping* function for the encoding. Due to the low capture rate of the Twitter API (only 1% of the tweets) and the non-uniform publishing behavior of the users, several values are missing in many popularity evolution sequences. Consequently, we decide to apply linear interpolation to fill small gap of maximum two missing values. Users with sequences presenting gaps of more than 2 symbols are discarded from our dataset. Table 4 presents the symbol distributions for the different datasets. We observe that non-popular accounts have as expected a zero or very moderate growth with 86.7% of C-symbol, *i.e.*, a gain or loss of maximum 5 followers. We note that they also have very few follower losses (0.6% of B-symbol which represents a loss of 5 to 100 followers), and are therefore very stable. For the popular accounts, the evolution is more varied: they can show significant growth, stagnate or have a significant decrease. Significant decreases can have several reasons: a sudden unpopularity due, for example, to a questionable decision-making, or else identification as a false account or an account having bought followers. Finally, accounts that become popular generally have rather long periods of significant growth, which explains our observation of around 49% of F, G and H symbols, so a gain of more than 100 followers.

Table 5: Number of patterns found with a min. support of 1%

pattern length	length 3	length 4	length 5
non-popular	11	18	27
popular	123	144	154
becoming popular	182	194	165
becoming popular - (non-popular + popular)	70	75	69

Traditional pattern extraction algorithms return item-set of symbols that are not necessarily closed. For our matching model, we need to extract only patterns that contains symbols which follow each others without gap. To execute a pattern extraction algorithm which take only frequents item-sets of consecutive symbols, we perform a pattern extraction on our 3 datasets using a sliding windows approach and reporting all sequences of k-symbols encountered. We decide to only keep patterns with a support greater than 1%. We report in Table 5 the number of patterns we found. We observe that the non-popular class is characterized by a low number of patterns. There are two reasons for this result: first, non-popular users exhibit a number of followers which does not vary in an important way, so most of the symbols for their popularity evolution are C or D symbols. Moreover, the class of non-popular users is very large, so a minimum support of 1% implies that this pattern is present in a very large number of popularity evolution sequences. With a support of 0.5%, the number of patterns is 286, 1051 and 2782 for respectively length 3, 4 and 5. The classes of popular and becoming popular users have approximately the same number of patterns, between 100 and 200. Observe that the size of the extracted patterns have a double impact on the number of patterns. Indeed, a pattern of k symbols with a support greater than 1% can provide potentially 2 or more patterns with $k+1$ symbols with a support greater than 1%. But oppositely, it can provide patterns with a support lower than 1%. This explains for instance why the number of patterns increases from 182 for size 3 to 194 for size 4 for the becoming popular dataset, and then decreases to 165 for size 5. When comparing the patterns found in the becoming-popular class to patterns from other classes, we see that several patterns are present in several classes. Nonetheless, we exhibit around 70 exclusive patterns that we will use for our detection.

We report in Figure 2 the support for the top-100 most frequent patterns that we extracted for each dataset. We first observe that regardless of the data set or the size of the extracted patterns, the frequency distribution of the patterns follows a power law. Patterns for the becoming-popular class have a more important support that those of other classes. For a size 3, the most frequent pattern is present among 55% of the sequences, the tenth most frequent among 15% and the fiftieth is still present in around 5% of the sequences. It results that the popularity evolution of becoming popular users is characterized by several dozens of patterns which can be used consequently to identify users from this class. The support for popular patterns is less important but remains significant: the most frequent pattern is present among 30% of the sequences, the tenth most frequent among 8% and the fiftieth is still present in around 3% of the sequences. For this class, we consequently observe the existence of a large number of characteristic patterns. However, as noticed in Table 4, the distribution of the symbols is less biased than the one of becoming popular users, what leads to more patterns but

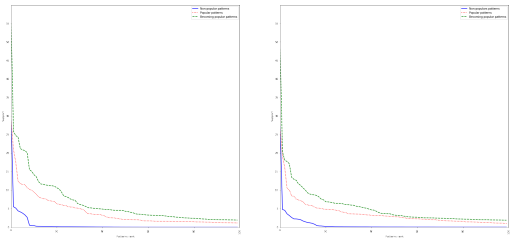


Figure 2: Support for 3 (left) and 4 (right) -symbol patterns

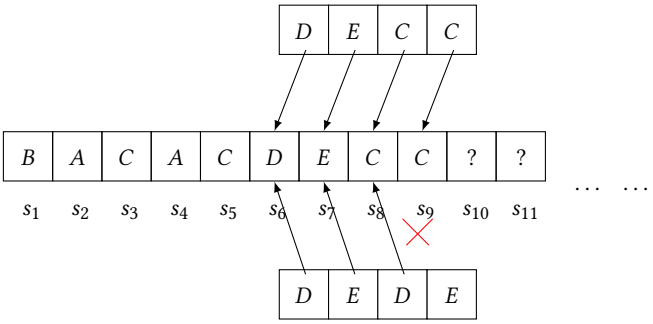


Figure 3: Example of matching attempt for patterns DECC and DEDE on a popularity evolution sequence

with less support. Finally the non-popular class is characterized by a small number of patterns with an important support: the most frequent pattern is present among 30% of the sequence, the tenth most frequent among 1%. Finally, we observe a long tail of patterns with a support lower than 1%. Two parameters explain this observation: first, the sequences of non-popular users are rather short since they have a low activity and we stop the evolution sequences if two or more symbols are missing, which is quite uncommon with popular or becoming popular users. Second, their number of followers is quite stable, as we can see in Table 4 with 86.7% of C-symbol. The results for a size 4 of pattern are similar, except we observe the power-law curves are a bit smooth compare to size 3. In fact, as explained above for Table 5, we have for the different classes fewer patterns with an important support but more with a medium support.

5 USING PATTERNS FOR AN EARLY DETECTION OF POPULAR USERS

Once we have identified the different sets of *exclusive* popularity patterns for the restrictions to the popular, non-popular and becoming popular evolution sets are denoted $\mathcal{S}_{\Pi^+}$, $\mathcal{S}_{\Pi^-}$ and $\mathcal{S}_{\Pi^*}$ respectively, we intend to use them to identify users becoming popular before they reach the popularity threshold ε.

5.1 Popularity pattern matching

Our objective is to test the matching of any pattern $p \in \mathcal{S}_{\Pi^*}$ with any popularity evolution sequence $s \in \mathcal{S}$ whenever it increases with a new symbol.

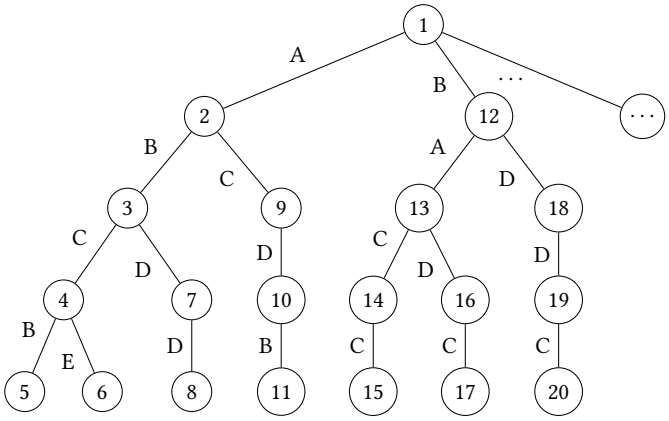


Figure 4: Tree structure for D_{pop} = {*ABCB*, *ABCE*, *ABDD*, *ACDB*, *BACC*, *BADC*, *BDDC*}

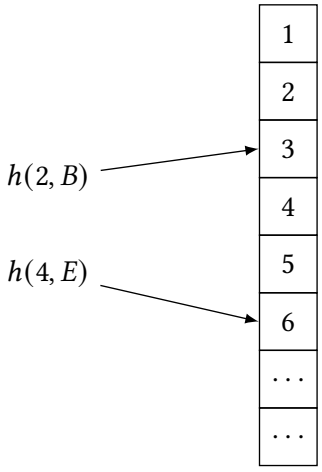


Figure 5: Our hash-based implementation

Example 1. *Figure 3 depicts an example of matching attempt. For our user, we report a new symbol C at week* 9 *corresponding to his popularity gain between weeks* 8 *and* 9*. We must then try to match each pattern of our whole set of* becoming popular *patterns with the suffice of the popularity evolution sequence. The pattern, DEDE does not match the suffice, oppositely to the pattern DECC. So we report a match and we consider the user corresponding to this popularity evolution sequence as a possible future popular user.*

So basically, our problem is a multi-stream multi-pattern matching issue. The difference with traditional pattern matching solutions (see Section 2) is that we have to deal with hundreds of millions of users and hundreds of patterns. This raises an important scalability issue. Our objective in the following is to propose a structure for a fast multi-pattern matching on a very large set of streams.

5.2 The H^2M index

To perform the matching of a set $\mathcal{S}_\Pi$ of patterns over a sequence π of symbols, several approaches are proposed in literature. Most efficient ones rely on finite state automaton (FSA). However, a traditional FSA presents some loops which results in testing for each state reached whether it corresponds to the final state of a given pattern or not. To limit this number of tests, we propose to rely on a trie representation, in other words a tree-shaped deterministic finite automaton. Since we make the assumption that our popularity patterns have a fixed size σ, it means that all (1) the paths from the root to a leaf have a length of σ and (2) only leaves correspond to the the final symbol of a popularity pattern.

Example 2. *Figure 4 is an example of the tree structure for the set of patterns { ABCB, ABCE, ABDD, ACDB, BACC, BADC, BDDC}. Here* $\sigma = 4$ *so we can check that all paths to a leaf have a length of* 4 *and each leaf corresponds to one pattern.*

This fixed-size of patterns allows us to consider a sliding window on the different sequences with only the σ last symbols which are used for the matching attempts.

Since we consider applications with hundreds of millions of users, we need to evaluate the transitions in a very efficient way. Thus we propose to choose the hash-based implementation for our pattern tree structure. So formally our pattern tree is defined thanks to our Pattern-Tree Hash index (PTH-index) as:

Definition 5. *[Pattern-Tree Hash index] A pattern-tree hash index PTH is defined on a pattern set $\mathcal{S}_\Pi$ as a couple (V, h_{trie}) where V is a set of nodes $v = (id, isLeaf) \in V$ with id a node id and isLeaf a boolean set to true when this is a leaf node, and $h_{trie} : V \times \Omega \rightarrow V$ is a hash function which represents the edges with the following properties:*

i) h_{trie} is injective, so each node could only have one parent (except the root node),
ii) if $\forall v \in V, v.isLeaf = true \Rightarrow \exists (x_1, x_2, \ldots, x_\sigma) \in \Omega^\sigma$, $h_{trie}(h_{trie}(\ldots(h_{trie}(root, x_1), x_2), \ldots, x_\sigma) = v \wedge x_1.x_2.\ldots.x_\sigma \in \mathcal{S}_\Pi$

Example 3. *Figure 5 represents the hash-based structure corresponding to the tree structure of the Figure 4. For instance with the symbol E, the node whose id is* 4 *leads to the node whose id is* 6*.*

For any incoming symbol for a given sequence π, we use this trie and we determine the new positions reached in this one. The following proposition determines the number of position in the trie we have to store for any user.

Proposition 1. *[Number of stored positions] Since the depth of the trie corresponds to the length of the patterns, σ, the different suffices with length in $[1, \sigma]$ must be considered when a new symbol is added and each suffice could reach a position in the trie. So at any time we have :*

- *to record for next matching attempt the different positions reached with the suffices with length in $[1, \sigma - 1]$,*
- *to report potentially a match when reaching a leaf Consequently the space requirement for storing user information is $O(\sigma)$.*

To efficiently retrieve the different positions in the trie for a given user, we propose to rely on a second hash structure. So consider the set $\mathcal{I}$ of the user identifiers. We then define our *Automaton Positioning Index* ($AP - index$) which consists of entries.

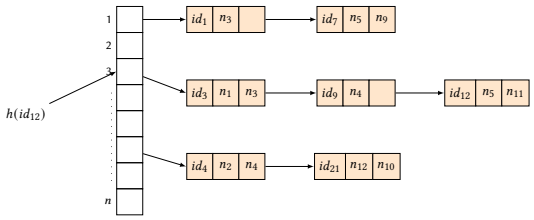


Figure 6: Our hash-based implementation

Definition 6. *Let $id \in \mathcal{I}$ a user identifier and V denotes the set of nodes ids from the trie. The entry indexing P, denoted $P(id)$, is a tuple (id, pos) with $pos \in 2^{V^\sigma}$ (the powerset of V^σ) a set of positions in the automaton.*

The indexing is "dense", *i.e.*, every popularity evolution sequence in the database is indexed, as soon as a user appears in the system, and by a different entry. $AP-Index$ is a hash file, denoted $AP[0..L-1]$, with directory length L (Fig. 6). Building a hash file with a hash entry for each user will lead to an extremely large index, since it requires $|\mathcal{S}|$ memory blocks and thus the index structure could not fit in memory. Consequently, the elements of AP refer to buckets or lines, each containing a list of entries. Each $AP[i]$ contains the address of the i-th bucket. Since entries have a similar size, *i.e.*, a user id and a set of positions with a number of elements between 1 and $\sigma-1$, we can store in a block with size B between $B/(|id|+(\sigma-1).|p|)$ and $B/(|id|+|p|)$ entries ($|id|$ and $|p|$ denote respectively the size of a user id and an automaton node id). So we set the size of the list accordingly. We consider in the following the pessimistic approach where all entries inside a bucket have $\sigma-1$ positions stored. We also assume we have $|\mathcal{S}|$ users to index. We explain below how to manage the dynamicity of the system, with users who join or leave frequently. With these settings, we can set the value of L:

$$L = |\mathcal{S}| \times \frac{|id| + (\sigma-1).|p|}{B}$$

Statistics over the entry size (so basically, what is the average number of automaton positions stored for a user) could permit to propose a L value with a higher space gain.

All together, the $AP-Index$ line structure is similar to a posting list in an inverted file. Since the key used for hashing are user ids, it is easy to design a hash function that evenly distributes the entries in the buckets, at least when the index is built. To manage the dynamicity of the users set, we could have several strategies which could moreover be combined. First, we could adopt the strategy used in database systems for storing the data, *i.e.* not to fill the data block (the *PCT-free* parameter). Thus, we could for instance choose a higher L value when creating the index and to have consequently some free space in each block to add new entries. In addition, since we use a hash file, lines should have a collision resolution method such as classical separate chaining that uses pointers to an overflow space. Such a technique accommodates moderate growth, but if we need to accommodate large growth, then we need a dynamic hashing method such as linear hashing [23].

The combination of our two hash-based structures for an efficient matching, $PTR-index$ and and $AP-index$, composes our proposal we name H^2M -index.

Table 6: Quality of the detection

	precision	recall	F1
Global dataset	0.7260	0.7604	0.7428
Removing popular users	0.9972	0.7604	0.8629

6 EXPERIMENTS

All experiments have been realised on a dedicated machine with 8x Intel® Xeon® Processor E7-4830 v2 (80 cores) and 512 Gb RAM. We have chosen Python for our development. To validate our approach, we first evaluate the quality of our detection, then we compare the performances of our matching structured with other implementations.

6.1 Quality of the detection approach

To estimate the quality of our approach, we split our global dataset of $32M$ users in a training set with 80% of the users, and a test set with the remaining 20%. We divide our training test into 3 groups of users according to their popularity evolution as explained in Section 4 and we perform our pattern extraction process on each of these user datasets. Tables 1, 2 and 5 describe the characteristics of these datasets along with their number of patterns.

Then, we try to match the patterns of the different classes on our test dataset to detect respectively non-popular, popular and becoming-popular users. Due to space limitation, we only present here the matching of becoming-popular patterns. We report results in Table 6. We observe that we have an overall precision of 0.7260. However, this precision can be largely improved by a fast pre or post processing by discarding popular users. In fact, we can at any time stop following a user that is identified as popular, *i.e.* those having at any time more than 4,000 followers. When discarding these users, we reach a precision of 0.9972. The recall reaches 0.7604, so a good $F1$ value of 0.8629. We achieve a higher recall when reducing the minimal support during the pattern selection. If we keep patterns with a lower support, we achieve a 0.85 recall, but the precision drops to 0.75 because these patterns are less characteristic of the associate class and consequently some sequences of other classes may also match.

Since our objective is to detect future popular users before they become popular, everytime a matching occurs, we measure the number of weeks between the week of matching and the week during which they actually become popular. We report the ratio of becoming popular users detected with respect to the number of weeks in advance that this user is detected actually popular in Figure 7. We observe that in 80% of cases, our approach allows us to detect a popular user at least 1 month before they actually become popular, and in 60% of cases, at least 2 months before. This rate is still around 40% for detection at least 3 months in advance. This confirms the interest of our approach and its effectiveness.

6.2 Scalabity

Finally, we perform experiments to study the scalability of our index and matching compared to existing matching structures. As competitors we implement:

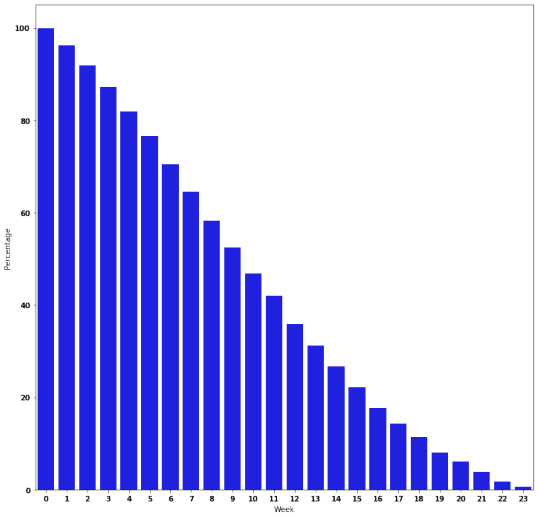


Figure 7: Number of weeks in advance a popular user is detected

- **SimpleTree**: implementation of a tree in standard python library with the possibility to search if a sub-sequence is in the tree. For each user we store in a hashmap the $\sigma - 1$ last symbol read (σ is the pattern size).
- **FSA**: we have use Automata-lib [2] which implements the structures and algorithms for finite automata.
- H^2M: our solution introduced above.

We generate according to the symbol distribution observed in our real dataset (see Table 4) datasets of respectively 1M, 10M and 50M popularity evolution sequences of 52 symbols (to simulate 1 year). We measure time to evaluate an incoming symbol for a user with the different structures along with the memory requirement to store information for the respectively 1M, 10M and 50M users. We report in Table 7 our results. We first observe that H^2M index allows to get the best performances for the matching since a new incoming symbol can be processed in $0.2ms$ when managing 1M users, so a gain of 90% and 250% with respectively *SimpleTree* and *FSA*. This matching time increases with the pattern size. We notice a 40% increase of the matching time for all structures. The rationale is that we have a 4-symbol subsequence in *SimpleTree* or *FSA*, or 3 positions in our structure, to consider instead of 3. Regarding the number of users to manage, we observe a constant matching time as expected, since we adopt a dynamic extension of the hashmap for the different structures to keep the current states for each users to avoid collisions. Consequently, the treatment of an incoming symbol is constant. We can also observe that our implementation requires a similar memory space compare to FSA while it saves 24% space compare to SimpleTree. The rationale is that we have to keep the $\sigma - 1$ last symbols for any user for the SimpleTree while we keep only 1 to $\sigma - 1$ reachable states for a user with H^2M. The memory gain increases with the pattern size, and we reach 44% gain for patterns with size 4.

[2] Automata-lib 3.1.0 : https://pypi.org/project/automata-lib/

Table 7: Performances of the matching

	Patterns size :	Size 3		Size 4	
Type	Nb of users	Time (ms)	RAM (MB)	Time (ms)	RAM (MB)
SimpleTree	1M	0.57	275.05	0.82	319.54
	10M	0.58	2526,00	0.81	3238.62
	50M	0.56	15067.56	0.80	16161.63
FSA	1M	1.05	205.86	1.53	227.55
	10M	1.05	2152.37	1.50	2199.18
	50M	1.03	11570.12	1.48	11962.11
H^2M	1M	0.30	214.97	0.41	222.32
	10M	0.30	2120.89	0.42	2197.58
	50M	0.31	11568.76	0.42	11954.67

7 CONCLUSION

This paper proposes a solution to tackle the early detection of future popular users. It is based on a characterization of users who become popular using popularity evolution patterns. Our analyzes show the existence of such patterns and our experiments confirm that they make it possible to detect future popular users several weeks before they are actually popular. We also offer a structure to allow scaling up of matching to millions of users.

We have several perspectives to complete this work. First of all we wish to make a more detailed analysis of the popularity evolution in order to determine the alphabet which will then allow a better detection. In addition, we want to look at other criteria (replies, quotes, likes, message propagation, etc.) and not only at simply the number of followers in order to identify influencers rather than popular users.

REFERENCES

[1] Klout score: Measuring influence across multiple social networks. In *IEEE Intl. Conf. on Big Data (Big Data)*, pages 2282–2289, 2015.
[2] Zeinab Abbassi, Aditya Bhaskara, and Vishal Misra. Optimizing Display Advertising in Online Social Networks. In Aldo Gangemi, Stefano Leonardi, and Alessandro Panconesi, editors, *Proc. Intl. Conf. on World Wide Web, (WWW) 2015*, pages 1–11. ACM, 2015.
[3] Mohamed Abouelhoda and Moustafa Ghanem. String mining in bioinformatics. In *Scientific Data Mining and Knowledge Discovery*, pages 207–247. Springer, 2009.
[4] Rakesh Agrawal, Tomasz Imieliundefinedski, and Arun Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.
[5] Çigdem Aslay, Wei Lu, Francesco Bonchi, Amit Goyal, and Laks V. S. Lakshmanan. Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret. *Proc. VLDB Endow.*, 8(7):822–833, 2015.
[6] Roja Bandari, Sitaram Asur, and Bernardo A Huberman. The pulse of news in social media: Forecasting popularity. In *Proc. Intl. AAAI Conf. on Weblogs and Social Media (ICWSM)*, 2012.
[7] Cody Buntain and Jennifer Golbeck. Automatically Identifying Fake News in Popular Twitter Threads. In *Proc. IEEE Intl. Conf. on Smart Cloud (SmartCloud)*, pages 208–215. IEEE Computer Society, 2017.
[8] Qi Cao, Huawei Shen, Jinhua Gao, Bingzheng Wei, and Xueqi Cheng. Popularity Prediction on Social Platforms with Coupled Graph Neural Networks. In *Proc. ACM Intl. Conf. on Web Search and Data Mining WSDM*, pages 70–78. ACM, 2020.
[9] David Dupuis, Cédric du Mouza, Nicolas Travers, and Gaël Chareyron. RTIM: A Real-Time Influence Maximization Strategy. In Reynold Cheng, Nikos Mamoulis, Yizhou Sun, and Xin Huang, editors, *Proc. Intl. Conf. on Web Information Systems Engineering (WISE)*.
[10] Manish Gaurav, Amit Srivastava, Anoop Kumar, and Scott Miller. Leveraging candidate popularity on twitter to predict election outcome. In *Proc. Intl. Work. on Social Network Mining and Analysis*, pages 1–8, 2013.
[11] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. Freespan: frequent pattern-projected sequential pattern mining. In *Proc. ACM Intl. Conf. on Knowledge Discovery and Data mining (KDD)*, pages 355–359, 2000.
[12] Jiawei Han, Jian Pei, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. Intl. Conf. on Data Engineering (ICDE)*,

pages 215–224, 2001.
[13] Jiawei Han, Jian Pei, and Yiwen Yin. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12, 2000.
[14] Liangjie Hong, Ovidiu Dan, and Brian D Davison. Predicting Popular Messages in Twitter. In *Proc. Intl. Conf. on World Wide Web (WWW)*, pages 57–58, 2011.
[15] Sounman Hong and Daniel Nadler. Which candidates do the public discuss online in an election campaign?: The use of social media by 2012 presidential candidates and its impact on candidate salience. *Government information quarterly*, 29(4):455–461, 2012.
[16] Imran N. Junejo and Zaher Al Aghbari. Using SAX representation for human action recognition. *Journal of Visual Communication and Image Representation*, 23(6):853–861, August 2012.
[17] Amir Karami and Aida Elkouri. Political Popularity Analysis in Social Media. *CoRR*, abs/1812.03258, 2018.
[18] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the Spread of Influence through a Social Network. *Theory of Computing*, 11:43, 2015.
[19] Paul Lagree, Olivier Cappe, Bogdan Cautis, and Silviu Maniu. Effective Large-Scale Online Influence Maximization. pages 937–942. IEEE, November 2017.
[20] Hui Li, Sourav S. Bhowmick, and Aixin Sun. *CASINO: Towards Conformity-aware Social Influence Analysis in Online Social Networks*.
[21] Hui Li, Sourav S. Bhowmick, and Aixin Sun. CINEMA: conformity-aware greedy algorithm for influence maximization in online social networks. page 323. ACM Press, 2013.
[22] Jessica Lin, Eamonn Keogh, Li Wei, and Stefano Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, August 2007.
[23] Witold Litwin. Linear Hashing: A New Tool for File and Table Addressing. In *Intl Conf. on Very Large Data Bases (VLDB)*, pages 212–223, 1980.
[24] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to Win! Predicting Message Propagation in Twitter. In *Proc. Intl. AAAI Conf. on Weblogs and Social Media (ICWSM)*, 2011.
[25] Ramakrishnan Srikant. *Fast algorithms for mining association rules and sequential patterns*. PhD thesis, Citeseer, 1996.
[26] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.
[27] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. TwitterRank: finding topic-sensitive influential twitterers. page 261. ACM Press, 2010.
[28] Yuto Yamaguchi, Tsubasa Takahashi, Toshiyuki Amagasa, and Hiroyuki Kitagawa. TURank: Twitter User Ranking Based on User-Tweet Graph Analysis. In Lei Chen, Peter Triantafillou, and Torsten Suel, editors, *Intl. Conf. on Web Information Systems Engineering (WISE)*, pages 240–253, 2010.
[29] Mohammed J Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.
[30] Tauhid R Zaman, Ralf Herbrich, Jurgen Van Gael, and David Stern. Predicting information spreading in twitter. In *Proc. Intl. Work. on Computational Cocial Science and the Wisdom of Crowds*, volume 104, pages 17599–601, 2010.
[31] Zilong Zhao, Jichang Zhao, Yukie Sano, Orr Levy, Hideki Takayasu, Misako Takayasu, Daqing Li, Junjie Wu, and Shlomo Havlin. Fake News Propagates Differently from Real News even at Early Stages of spreading. *EPJ Data Sci.*, 9(1):7, 2020.

DRLindex: Deep Reinforcement Learning Index Advisor for a Cluster Database

Zahra Sadri
School of Computer Science
University of Oklahoma
Norman OK USA
Zahra.sadri@ou.edu

Le Gruenwald
School of Computer Science
University of Oklahoma
Norman OK USA
ggruenwald@ou.edu

Eleazar Lead
School of Computer Science
University of Minnesota Duluth
Duluth MN USA
eleal@d.umn.edu

ABSTRACT

Cloud database providers provision different architectures to guarantee high availability. One of these architectures is a cluster database that consists of several database engine nodes, where data is replicated among the nodes. Although the cloud database providers provide various auto-indexing tools, these tools mostly address characteristics of a database deployed on a single node, not a cluster. It is possible to install an index advisor on each node, which recommends an index set for that node. The problem with this approach is that the current index advisors for a single node aim to minimize the processing cost of the workload; however, on a cluster database, other goals such as load balancing can be considered. Hence, the better solution could be an index advisor which has a comprehensive view of the cluster node.

In this paper, we propose an index advisor for a replicated database on a database cluster for a read-only workload. The advisor considers both query processing cost and load balancing. It utilizes a Deep Reinforcement Learning (DRL) approach in which a DRL agent learns to select a set of index configurations for nodes in a cluster. We describe the components of the DRL-advisor such as the agent, the environment, a set of actions, the reward function, and other modules. Experimental results validate the effectiveness of the algorithm.

CCS CONCEPTS

• Insert Information systems ~Data management systems ~Database administration ~Autonomous database administration

KEYWORDS

Index tuning, Cluster database, Deep reinforcement learning

ACM Reference format:
Zahra Sadri, Le Gruenwald and Eleazar Leal. 2020. DRLIndex: Deep Reinforcement Learning Index Advisor for a Cluster Database. In *Proceedings of ACM IDEAS conference (IDEAS'20), August 12-14, Seoul, Republic of Korea, 8 pages.* https://doi.org/10.1145/3410566.3410603

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410603

1 Introduction

Cloud database providers such as Microsoft Azure SQL Database provide different architectures to guarantee high availability and reduce the impact of failures. One of those architectures can consist of a primary database and one or more secondary databases. Data is fully replicated to secondary databases. The primary database handles read-write workloads, while secondary databases are only used for processing read-only workloads. Using this architecture, read-only queries can be distributed and processed by secondary replicas. Selecting and creating index sets for replicas can substantially reduce the workload processing cost.

Index tuning is the task of selecting and creating an index configuration for a workload with the aim of minimizing the processing cost of the workload via reducing the I/O. An index configuration is a set of indexes defined over the tables in the database. Index tuning is a complicated task even for just one node as different factors impact the index selection such as the workload and data distribution in the database. Usually, a database administrator (DBA) is responsible for this task. To relieve the burden of this complicated task, many index-advisor tools have been proposed [1] [3] ,[4],[6]-[8] over decades. The main issue to overcome is the large search space of the combinations of promising attributes for indexing. The aim is to find the combination of attributes that minimizes the processing cost of the workload. For large databases, enumerating all possible combinations is impossible. Therefore, the search space needs to be pruned such that we do not miss useful configurations. Various heuristic and optimization algorithms are applied to achieve effective search solutions that are near optimal. These methods are based on assumptions that might cause these index advisors to miss some good configurations or select index configurations that cause regression in the query processing performance[8]. Hence, a Database Administrator (DBA) cannot rely on these tools completely and sometimes needs to select indexes in a trial and error approach.

For a centralized database on a single node, an index advisor selects only one optimal (or near-optimal) index configuration. When it comes to a cluster database consisting of a number of replicated databases (which we call replicas in the rest of the paper) for processing the workload, a question that arises is whether to create the same index configuration on all replicas

known as Uniform Index Configuration (UIC) [6], or different index configurations known as Divergent Design [6][18]. Divergent Design [6][18] aims to create a different index configuration for each replica by specializing each replica to a subset of the workload with the aim of minimizing the workload processing cost. It introduces a load-balance parameter l that defines the number of specialized replicas with proper index configuration for processing a query with a low cost. One problem with the divergent design [6] is creating different loads on replicas. That is, one replica might get overloaded while others are almost idle.

Recently, advances in machine learning, especially in deep neural networks and reinforcement learning, encourage researchers to move toward the learner advisors [2][8][10][13]-[15] instead of heuristic approaches. Motivated by the recent works and shortages in the current designs, this paper introduces a Deep Reinforcement Learning (DRL)-index advisor for selecting index configurations for fully replicated databases in a cluster. The agent will learn to choose an index configuration for each replica. The DRL-agent employs a deep neural network in its learning process and receives a reward based on the processing cost and load balancing. The main contribution is to consider the load balancing besides the processing cost to avoid the load skew in a cluster database, i.e., cases where some nodes either are idle or process a very low workload while other nodes are overloaded.

The remainder of the paper is organized as follows: In section 2, we briefly introduce the background on RL/DRL. We introduce the architecture and the components of our proposed index advisor, called DRL index advisor (DRLinda), in Section 3. Section 4 describes the experiments conducted to evaluate the performance of DRLinda. Section 5 reviews the literature in the areas of index tuning. Lastly, we conclude the paper and discuss future research directions in Section 6.

2 Background

We review the concept of Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL). RL deals with a sequential decision-making problem and is helpful when there is no supervised feedback available, instead evaluative feedback is available [17]. Contrary to supervised learning, there is no label available, and the evaluative feedback conducts the learning process. RL can be applied to the problems that can be modeled as a Markov Decision Process (MDP). A Markov Process consists of a set of states and transition probabilities that are stochastic. The transition among states only depends on the prior state.

2.1 Reinforcement Learning

RL consists of several components including an agent, an environment, a policy, a reward function, and a value function which are defined as follows [17]:

- Agent: the learner part of RL that interacts with an environment.
- Environment: whatever the agent interacts with. An environment can be defined by its components such as a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, and a reward function. The state of the environment transitions from a current state s_i to the next state s_{i+1} after applying action a_i.
- Reward Function: a method for determining the objective of the RL problem. It evaluates the action and returns a value called reward rw.
- Policy π: an agent takes an action in a state based on the policy. In simple words, policy guides an agent to take which action at each state.
- Value Function: The quality of a policy is quantified by a value function that associates to each state an expected cumulative discounted reward. Discounted value γ implies the impact of the future rewards.

Put all these components together, in a sequence of steps or an episode, at any time step t, the agent takes an action a_t based on a policy π and the state of the environment changes from the state s_t to the new state s_{t+1}. The agent receives the reward value rw_t from the environment for the selected action a_t. The objective of the agent is to find the optimal policy that governs the agent to maximize the cumulative expected rewards known as a return in the long term.

In some cases, the state transitions and rewards are unknown and only a set of states and actions are available. For solving such cases, one of the possible methods is a model-free and off-policy algorithm called Q-learning [17]. This method evaluates the value of each action a in a state s using a Q-function. The goal is to select an optimal value in each state that reflects the most rewarding action. The agent improves its behavior through learning from the history of interactions with its environment. At each time step t, the Q-function approximates the value of each action in the long term using the Bellman equation [13] as follows:

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\,[rw_{t+1} + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (1)$$

When the number of states and actions are limited, Q-learning can use a look-up table known as Q-table to keep the value of each action at each state. However, for a large number of states and actions, Q-table is not an applicable approach. Instead, a value function approximator such as a neural network is used to estimate the value of actions in states. This method combines the RL algorithm with a deep learning algorithm called Deep Reinforcement Learning (DRL) [12]. Likewise, DQN is a Q-learning approach that utilizes a neural network $Q_\theta(s, a)$ with weights θ in the learning process of its agent [12]. The neural network helps the DRL-agent to predict the values of actions at the output layer of the neural network. Weights in the neural networks are updated by Gradient Decent on the loss $L(\theta)$:

$$\mathrm{L}(\theta) = (rw_t + \gamma max_{a_{t+1}} Q_\theta(s_{t+1}, a_{t+1}) - Q_\theta(s_t, a_t))^2 \quad (2)$$

This loss is the squared difference between the target (observed reward r_t plus the discounted estimate of future returns of new state s_{t+1}) and the current estimate of $Q_\theta(s_t, a_t)$.

2.2 Training

The data within the steps of one training episode can be strongly correlated. However, a neural network needs to learn from less correlated data. To address this problem, the experience replay approach [12] is proposed which stores the samples of agent's experiences in the form of $(s_t, a_t, rw_t, s_{t+1})$ in a buffer. These samples are selected randomly and used to train the agent.

2.3 Exploration-exploitation dilemma

An agent needs to explore the possible actions in various states to learn about actions which lead to obtaining more rewards. On the other hand, the agent needs to exploit its experiences and takes actions that it has already tried and known by taking those actions, it can gain considerable rewards. The dilemma is balancing between the exploration and exploitation, i.e., keep trying new actions and take advantage of the best ones. The problem arises when the agent keeps selecting the best action and not trying to learn more possible beneficial actions. One approach to address this problem is ε-greedy policy, which the value of ε defines the probability of taking a random action and is between 0 and 1[17]. When we want an agent solely to explore, ε can be set to one. Overtime that agent learns about actions, the value of ε can be gradually decreased. If we set ε to zero, the agent only exploits its experiences and will not try new actions.

3 DRL-BASED INDEX SELECTION FRAMEWORK

This section discusses our framework for index selection for a cluster database. First, we provide an overview of our solution. Then, we explain the core elements of the RL approach, i.e., the environment, and the design of a DRL-based agent.

3.1 Overview

We design a DRL-index advisor that learns how to select index configurations for different replicas on a cluster database. The recommended set of index configurations can be either the same index configuration for all replicas (uniform design), or different index configurations for different replicas in a cluster for a specific part of the workload (divergent design). When agent recommends a divergent design, it provides a routing table that determines which replica has a better index for processing a given query. It is possible to install one of the existing index advisors on each node, and each node selects its index configuration. However, our insight is that a centralized index advisor with a comprehensive view of the workload can decide better than the former approach.

As shown in Figure 1, our solution consists of several modules. We assume the workload is known in advance using a state-of-the-art workload predictor [11]. The workload predictor module receives a workload and a time interval. The predictor has an internal database to store executed queries in the cluster database. First, this module extracts the query templates, and stores the total

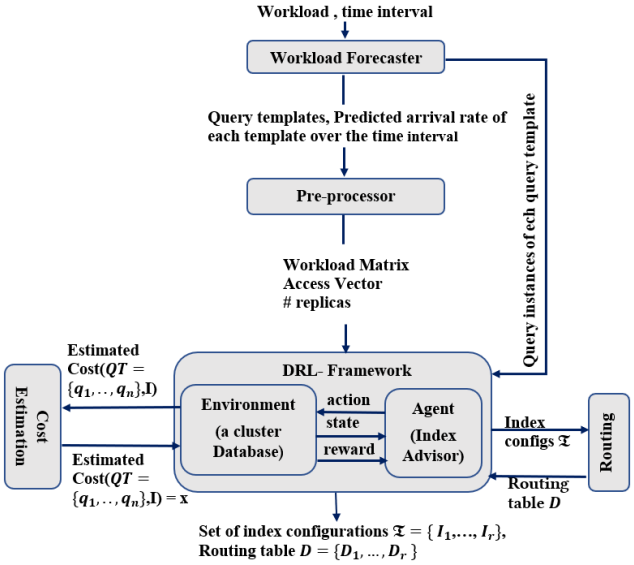


Figure1. The DRLinda procedure for a cluster database

number of query instances per template known as total count per template [11]. A query template is a representative of a group of of query instances that they have the same format but might have different parameter values. For instance, suppose we have two query instances; SELECT * FROM T1 WHERE T_1.att1> 10, and SELECT * FROM T1 WHERE T_1.att1> 100. Here, SELECT * FROM T_1 WHERE T_1.att1 > # is a query template of these two instances, where the value of # can change in different query instances. Second, the predictor clusters similar query templates based on their arrival rates, and eventually, it predicts the arrival rate of each cluster [11]. This module provides the query templates, the query instances of the query templates, and the total count per query template. The query templates and the total count per query template are inputs to the next module. The query instances are used to compute the costs of query templates in the cost calculation module.

The pre-processing module extracts the characteristics of the workload. It uses the query templates to extract the attributes which are appeared in their Where or Join clauses known as indexable attributes. Then, it creates a matrix of size $k \times m$ where k is the number of query templates and m is the number of indexable attributes. An entry is set to one if the indexable attribute appeared in the Where or Join clause of a query template; otherwise, it is set to zero. Moreover, using the total count per query template, it creates an access vector that shows the total number of query accesses to each indexable attribute. Inputs to the DRL framework module are the workload matrix, access vector, and the number of replicas.

The DRL framework formulates the index selection problem for a cluster database as a DRL problem. It works based on the RL concept as described in Section 2, where the agent interacts with the environment. Similarly, in our case, the agent learns to select a set of index configurations by interacting with the replicas in the cluster. As the number of actions and states are large, we use

a deep neural network in the learning process of the agent. The DRL agent follows the off-policy deep Q-learning algorithm [17]. The agent decides which attributes are to be indexed for each replica in a cluster. Then, each replica in a cluster applies that action to get the new state and computes the reward. Our reward function is computed based on two objectives: reducing both the processing cost of the workload and the load-skew. The reward value is a weighted sum of these two objectives. To compute this reward, the environment interacts with the cost calculation and routing modules. These modules provide the estimated processing cost of the query workload and a routing table, respectively. Then, the environment calculates the total processing cost of the workload and the amount of workload skew on each replica. The weighted sum of these two values is the final reward value. We explain each of these modules in detail later. The final output is a set of index configurations denoted as $\mathfrak{T} = \{I_1, \dots, I_r\}$ and a routing table denoted as $\mathcal{R} = \{R_1, \dots, R_r\}$. Where I_i is the index configuration for replica $i \in [1, r]$ and R_i is the set of queries which should be routed to replica $i \in [1, r]$.

3.2 Cluster Database (Environment)

Here, we define the components of the RL environment including states, actions, and reward function, respectively.

States Representation. The current set of index configurations in the cluster database and workload characteristics present the state of the cluster. We represent the current index configurations of replicas by a matrix of size $r \times m$, where r is the number of replicas and m is the number of indexable attributes, and we call this matrix state matrix. An entry in state matrix is set to 1 if there is an index on a replica $i \in [1, r]$ and indexable attribute $j \in [1, a]$; otherwise it is set to 0. For instance, suppose we have two replicas r_1 and r_2 and two indexable attributes a_1 and a_2 where the first replica has an index on attribute a_2 while the second replica has no index, then state matrix will be the following:

State matrix =

	a_1	a_2
r_1	0	1
r_2	0	0

The state matrix, the workload matrix and the access vector are used to show the state of the environment.

Set of Actions. The set of possible actions $\mathcal{A}$ includes:

- Creating an index on a specific indexable attribute on a particular replica.

Our solution is based on the episodic MDP model [17]. Using this model, our solution starts with an initial state s_0 where there is no index on replicas and ends at a terminate state s_t after selecting a specific number of indexes on each replica, or reaching to the specific disk space budget, or a timeout, or combinations of them. In this paper, we consider the maximum number of indexes $\mathcal{N}$ as the constraint which is a user-defined parameter.

Reward Function. We design a reward function with two main objectives: 1. Workload processing cost: a scalar reward is given with respect to improvement in the estimated processing cost of the workload in the presence of the recommended index configurations; and 2. Workload Skew: a reward value is assigned based on the total workload skew in the cluster database. The less the workload skew, the more the reward. In the following, we describe how to measure the reward of each objective.

Workload processing cost. The objective is to choose a set of index configurations for a cluster database such that minimizes the total estimated processing cost of the workload. To obtain the minimum processing cost; the query instances of each query template must be processed by a replica that has the best index configuration for that template. That is a replica which minimizes the processing cost of the queries of that template. The problem with this approach is that the best replica might not be available for any reason. To avoid this problem, we consider more than one replica for processing the query instances of each template. We explain the way of selecting each replica in the routing module in details. The routing table identifies the replica(s) which can process a query template based on the index configuration.

First, we compute the cost of each query template using Equation 3:

$$TemplateCost\ (QT, \mathfrak{T}) = \ \sum_{\mathrm{q}\in\mathrm{QT}} \sum_{\mathrm{R}\in[1,n]} cost(\mathrm{QT}, I_R)/n \quad (3)$$

Where q shows the query instances of a query template, n denotes the number of replicas defined in the routing table for processing a specific query template. The value of n can vary between 1 to r. Also, I_R is the index configuration of the selected replica. Then, we accumulate the cost of all query templates in the workload as below:

$$TotalCost\ (W, \mathfrak{T}) = \sum_{\mathrm{QT}\in W} TemplateCost\ (QT, \mathfrak{T}) \quad (4)$$

The reward value is computed as follows:

$$reward(\mathfrak{T}) = \frac{TotalCost(W,\emptyset) - TotalCost(W,\mathfrak{T})}{TotalCost(W,\emptyset)} * 100 \quad (5)$$

Where $\boldsymbol{TotalCost(W, \emptyset)}$ denotes the estimated processing cost of the workload when there is no index configuration on replicas.

Workload-Skew. We want to minimize the workload skew among the replicas when recommending index configurations. We define workload skew as a considerable difference between the amount of the workload that each replica processes. For instance, suppose in a database cluster with three nodes, one node is completely idle, the other one processes a small part of the workload while the third one is almost overloaded by processing the large portion of the workload. We want to avoid these kinds of situations. The ideal is to have the workload distributed evenly among all replicas. In this case, the workload skew is considered to be zero. It is almost infeasible to achieve zero skew; therefore, we want to decrease it as much as possible. We define a factor as workload-skew to mitigate this issue. The goal is to create index configurations for replicas in a way that not only minimizes the

workload processing cost but also balances the workload among the replicas as much as possible.

To compute the workload-skew, first, we compute the amount of workload that each replica should process in the best case, i.e., when the workload is evenly distributed among replicas as follows:

$$workload\ (R)_{bestcase} = \frac{TotalCost\ (W,\mathfrak{I})}{r} \quad (6)$$

Second, we calculate the amount of the workload that each replica R with index configuration I_R should process:

$$workload\ (R)_{real} = \sum_{QT \wedge R} cost(QT, I_R) \quad (7)$$

In Equation (7), $QT \wedge R$ means the query templates the query instances of which are processed on replica R. Third, we compute the workload skew on each replica as below:

$$workload\_skew\ (R) = \frac{|\text{Workload (R)}_{real} - \text{Workload (R)}_{bestcase}|}{\text{Workload (R)}_{bestcase}} \quad (8)$$

In the above equation, the value of workload-skew (R) either greater or less than zero implies that the replica suffers from over skew or under skew, respectively. We want to reduce these cases. The value of zero shows that there is no skew on a replica, which is the desirable case. After finding the amount of the workload skew on each replica, we calculate the sum of the workload-skew values of replicas. Finally, the reward of the workload skew is computed using Equation 9:

$$reward(S) = \frac{1}{\sum_{R \in r} workload\_skew(R)} \quad (9)$$

Final reward. Eventually, the reward is a weighted sum of $reward(\mathfrak{I})$ and $reward(S)$ as follows:

$$\alpha \times reward(\mathfrak{I}) + \beta \times reward\ (S) \quad (10)$$

In Equation 10, α and β are obtained by trial and error and their values defines a trade-off between cost reduction and load-balancing.

3.3 DRL-Agent

The agent is in charge of learning the specific sequence of actions for all replicas in the cluster to maximize the reward function. It uses Deep Q-Learning [17] to predict the next action a. The workload matrix, access vector, number of replicas, and state matrix are inputs to the neural network. These inputs are provided to guide the agent in finding the nonlinear relations between indexes. For instance, the workload matrix shows which query templates access similar indexable attributes. This can help the agent to group query templates based on their accesses to indexable attributes and creates an index for each group of query templates on different replicas. The access vector shows which attributes are accessed more frequently and encourages the agent to create more numbers of indexes on them.

DRLindex Advisor
Inputs: Workload W; Query templates QT;
Query instances of each template Q_{qt};
number of queries per template $count_{Q_{qt}}$;
number of replicas r; number of episodes e;
number of timesteps T; number of samples of minibatch σ, capacity p
set of actions $\mathcal{A}$,
Outputs: trained model to recommend index configuration for replicas
Function PreProcessing (QT, $count_{Q_{qt}}$, r)
1: CIS = [] //indexable columns
2: **foreach** $qt \in QT$ do
3: (CIS) ← Extract attributes appeared in the template qt
4: **end for**
5: WM ← create the workload matrix
6: AV ← create the access vector
7: **return** WM, AV
Function DRLIndexAdvisor (CIS, WM, AV, Q_{qt})
8: Randomly initialize Q-network Q_θ
9: Randomly initialize Q-network $Q_{\theta'}$
10: Initialize replay memory D to capacity P
11: Initialize action space $\mathcal{A}$ with CIS
//set the initial index configuration to no index for all replicas
12: IC ← Initialize Index configuration matrix
13: Initialize observation state (state matrix) = WM ∪ AV ∪ IC
14: **foreach** $i \in [1, e]$ do
15: state s_0 ← initial state matrix //reset state
16: **foreach** $t \in [1, T]$ do
17: with probability ε select a random action a_t from $\mathcal{A}$
18: otherwise $a_t = argmax_a Q_\theta(s_t, a)$
19: s_{t+1} ← Execute action and update the state matrix to show the new index configuration for each replica
// see reward function in the section 3.2
20: rw_t ← Compute the reward
21: Store transition (s_t; a_t; rw_t, s_{t+1}) in M
22: sample experiences ← get_random_sample (σ, D)
23: Train Q-network with Stochastic Gradient Decent and loss
24: Update weights of target model every z steps $Q_{\theta'} = Q_\theta$
25: **end for**
26: **end for**
27: **end function**

Figure2. Pseudocode for DRL-Index Advisor

The agent acts in a predefined number of episodes. An episode starts with the initial state s_0 where there is no index on replicas. At each time step, the agent selects an index for each replica in the cluster database. The agent might select the same index or a different index for each replica. In our work, an episode ends after the agent selects a specific number of indexes $\mathcal{N}$ for each replica. This value can be the same or vary per replica.

3.4 The Cost Calculation Module

This module receives the query instances of each template and the recommended index configuration for a replica as inputs. Then, it estimates the processing cost of the query instances in the presence of a new index configuration using the What-if optimizer tool [5]. Since each node has a query optimizer, the cost calculation can be done in parallel on all replicas. To compute the cost of a query template, first, the query optimizer provides the estimated cost of each query instance of the template. The cost of a query template is the sum of the cost of the query instances. The output of this module is the total and average cost of a query template.

3.5 The Routing Module

This module receives the set of the selected index configurations as the input and provides a routing table as the output. The routing table ranks the replicas from the best to the worst with the best being the replica which has the best index configuration for a query template, i.e., this replica reduces the estimated processing cost of the template the most. We consider this method as the best routing policy. However, the best replica might be unavailable for any reason; therefore, the routing table keeps not only the best replica but also those that have acceptable processing cost for a query template. In the following, we explain what processing cost considers acceptable in our view.

To rank the replicas, the routing module computes the average processing cost of each template on each replica. To calculate the processing cost of each template, it calls cost estimation modules. Then, it computes the average estimated cost of each template in the cluster database. Next, it compares the average processing cost of each template on each replica with the average processing cost of that template in the cluster. Finally, for each template, it adds only those replicas that have the processing cost less than or equal to the average processing cost of the template in the cluster database in a sorted way. The routing modules returns a routing table that represents the proper replicas for processing each query template. Notice that the number of the proper replicas can vary per query template. We call this approach the average routing policy. The pseudocode of the DRLinda is depicted in Figure 2.

4 Experimental Results

This section presents the results of an experimental study conducted to evaluate the performance of DRLinda. We first describe the experimental methodology and then discuss the results.

4.1 Methodology

Dataset and Workload. We use the database schema and query workload from the standard TPC-H benchmark [19]. Its schema consists of 8 tables and the query workload includes 22 query templates. We loaded a 1GB database with the TPC-H data and generated the query workloads using the provided query generator. The training workload consist of 400 queries of 12 templates out of 22 templates which templates are selected randomly. Also, we generated 65 queries to test the trained agent.

Experimental Setups. Our experiments use an implementation of the DRL-index advisor written in Python which employs neural networks implemented in Keras and TensorFlow works as its backend. The neural network is the function approximator with 2 hidden layers each with 128 neurons. The activation function for the hidden layers and the output layer are RELU and Linear, respectively. Learning is executed using an adaptive moment optimizer (Adam) with a learning rate of 0.001.

The database system in our experiments is PostgreSQL. For the testing platform, we created cluster database with two replicas in CloudLab [20], a scientific infrastructure for research on cloud computing. The advisor and PostgreSQL run on Ubuntu Linux

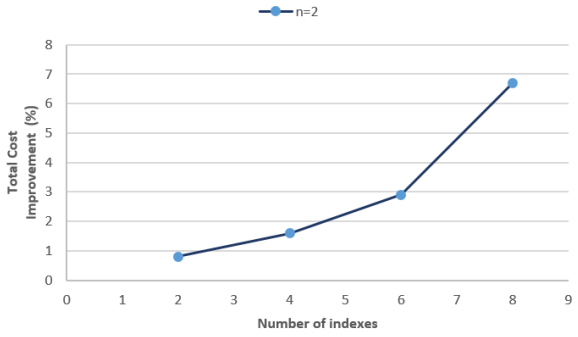


Figure 3. Performance of DRLinda for a cluster with 2 replicas.

18.04 LTS with 128GB of DDR4 main memory, two Intel Xeon Silver 4114 10-core CPUs, and a 10Gbps interconnect.

Baseline. We compare the recommended set of index configurations produced by our algorithm DRLinda $\mathfrak{T}_{drl}$ with the baseline where all replicas have the same index configuration called uniform design $\mathfrak{T}_{uni}$. To find the uniform design, we use POWA [21] the index advisor of PostgreSQL. We provide our workload set to POWA and create its recommended index configuration for the workload on all replicas.

Experimental Dynamic Parameters. We study the impact of changing the number of created indexes on the query performance by varying their values in the experiment.

Metrics. We measure the performance of the studied index advisors using the total query processing cost, i.e., the total cost to process all queries. Total processing cost is estimated by the query optimizer. Moreover, we created the recommended index configurations and report the average response time of the workload. Query response time is a time that system executes the query. To compute the average response time, we repeat the experiments in five independent runs and computes the average.

4.2 Results

The first experiment evaluates the performance improvement of DRLinda in comparison to the baseline. The second set reports the performance of the recommended $\mathfrak{T}_{drl}$ by DRLinda based on the total processing cost and average execution time in comparison with uniform index configuration $\mathfrak{T}_{uni}$ as we change the number of created indexes $\mathcal{N}$. We consider the same number of indexes for all replicas, i.e., we created $\mathcal{N}$ index on each replica.

Performance of DRLinda. Figure 3 depicts the improvement of DRLinda in terms of the total processing cost over the uniform design. For a small number of indexes, the improvement of $\mathfrak{T}_{drl}$ is not considerable. The reason is that the recommended index set by DRLinda $\mathfrak{T}_{drl}$ is almost similar to $\mathfrak{T}_{uni}$. As we increase the number of indexes, $\mathfrak{T}_{drl}$ shows a significant improvement over $\mathfrak{T}_{uni}$. The reason is that, $\mathfrak{T}_{drl}$ suggests more diverse index configurations compare to $\mathfrak{T}_{uni}$ which are more specific to the part of the workload.

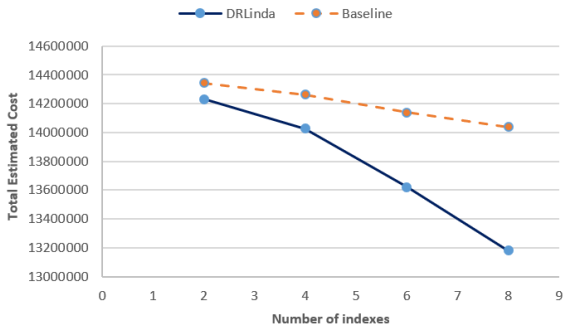


Figure4. Total processing cost of the workload for a cluster with two replicas

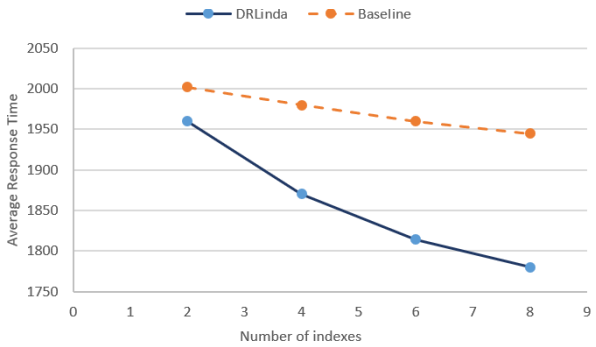


Figure5. Average response time of the workload for a cluster with two replicas

In overall our approach shows a considerable improvement even with creating a small number of indexes over the baseline. Figure 4 shows in detail the how the total processing cost changes as we vary the number of indexes. At the beginning, both approaches show a very similar behavior and not a notable deduction in the total processing cost of the workload. Later, as the number of indexes is increased, $\mathfrak{T}_{drl}$ performs better than the uniform design and reduces the total processing cost of the workload remarkably.

Figure 5 illustrates the average response time of executing the workload. We install the recommended index configurations by DRLinda and uniform design. Then, we execute query workload five times independently and compute the average response time. $\mathfrak{T}_{drl}$ reduces the response time notably in comparison with $\mathfrak{T}_{uni}$. The reason is that based on the routing suggested by DRLinda, the queries are executed by the replicas that have the best index configuration for those queries.

5 Related Work

Index tuning has been studied since the '70s and continues to be an active area of research. Different index advisors have been proposed based on the workload characteristics over the decades. In the following, we discuss existing auto-indexing algorithms.

Offline Auto-Index Advisors: Offline index tuning methods [1],[4],[7],[8]] assume that the workload is static and known in advance. Due to this assumption, the auto-indexing algorithm suggests the index configuration only once. These algorithms are called offline because they run in offline mode, i.e., all the computations are performed before executing the workload. These algorithms only suggest the index configuration; they do not create indices automatically. Therefore, a DBA should decide whether to install the recommended index configuration or not. After installing the index configuration, the workload will be executed. Offline algorithms require human interactions for providing the sample workload, installing the recommended index configuration, deciding when the tuning session is required, and when to deploy the recommendations. All these tasks are non-trivial.

Online Auto-Index Advisors: These algorithms [2],[16] are emerged to reduce human interaction and deal with the workload changes in a dynamic environment. On contrary to Offline algorithms, the workload is considered dynamic and not known completely. These algorithms are considered online index configuration selection is done in parallel with query execution. To detect the workload changes, the algorithm monitors the workload continuously, creates the selected index configuration, reevaluates the index configuration periodically, and updates the configuration based on the changes in the workload or the database. These algorithms assume that the coming workload will be similar to the recently executed queries. However, this assumption is not always valid. Also, online algorithms decide to update the index configuration after an interval of observing and finding enough evidence that proves the efficiency of the new index configuration. By the time of its decision, if the workload changes, its recommendation is not valid anymore.

Auto-Index Advisors for a Replicated Database: We discussed auto-indexing algorithms for a centralized database on a single node. There are a couple of works for a replicated database [6],[18]. The main objective of these tools is to employ replication for tuning the database system. DivDesign [6] proposes the concept of divergent design to specialize each replica for a subset of a workload with a proper index configuration. This design reduces the overhead of index maintenance by decreasing the number of indexes on each replica. This heuristic has a few limitations. First, this method ignores the changes in the workload and the probability of a failure of a replica. Also, it suffers from creating an uneven load on replicas. RITA [18] addressed these limitations and formulated a divergent design problem as a Binary Integer Problem (BIP). Existing BIP solvers are utilized to solve this optimization problem. Both DivDesign [6] and RITA [18] use a load-balance parameter to define the number of replicas that are specialized to process a query type. The problem is that they assume the same load-balance parameter for all query types. The problem with this assumption is that the frequencies of different query types vary, and some might take more time to be processed.

AI-Index tuning Advisors: Some auto-indexing methods employ one of the existing machine learning or data mining algorithms in their solutions. Recently, RL/DRL algorithms attract researchers' attention. One of the early works utilizes RL to learn the cost model [2]. The others apply different DRL algorithms for index selection [10],[15]. These are all based on the query cost

estimated by a query optimizer and for a centralized database. Our work also uses DRL for index selection but for a cluster database where our algorithm trains the agent to choose the index configuration for each replica in the cluster. Moreover, our solution considers the load balancing issue which exists in heuristic algorithms, such as DivDesign [6].

6 Conclusion and Future Research

In this paper, we introduced a new approach for a learning index advisor to identify index configurations for replicas in a cluster database using DRL that not only reduces query processing cost but also improves load balancing. The main idea is that a DRL agent learns its decisions based on the experiences by monitoring the rewards via trying different index configuration. We showed that our approach can select well-performed index configurations via experiments.

In the future, we plan to conduct additional experiments with workloads that are more diversified and include updates in the cluster databases with more numbers of replicas. Moreover, we want to extend our current design to a multi-agent design and compare these two designs together.

REFERENCES

[1] Sanjay Agrawal, Surajit Chaudhuri, Lubor Kollár, Arunprasad P. Marathe, Vivek R. Narasayya, and Manoj Syamala. 2004. Database Tuning Advisor for Microsoft SQL Server 2005. In VLDB. 1110–1121.

[2] Debabrota Basu, Qian Lin, Weidong Chen, Hoang Tam Vo, Zihong Yuan, Pierre Senellart, and Stéphane Bressan. 2015. Cost-model oblivious database tuning with reinforcement learning. In *Database and Expert Systems Applications*. Springer. 253-268.

[3] Nicolas Bruno and Surajit Chaudhuri. 2007. An Online Approach to Physical Design Tuning. In *Proceedings of the IEEE 23rd International Conference on Data Engineering (ICDE)*, 826–835.

[4] Surajit Chaudhuri and Vivek R. Narasayya. 1997. An efficient, cost-driven index selection tool for Microsoft SQL server. In *VLDB*. 146-155.

[5] Surajit Chaudhuri and Vivek R. Narasayya. AutoAdmin. 1998. 'What-if' Index Analysis Utility. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. 367–378.

[6] Mariano P. Consens, Kleoni Ioannidou, Jeff LeFevre, and Neoklis Polyzotis. 2012. Divergent physical design tuning for replicated databases. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. 49-60.

[7] Debabrata Dash, Neoklis Polyzotis, and Anastasia Ailamaki. 2011. CoPhy: A Scalable, Portable, and Interactive Index Advisor for Large Workloads. PVLDB 4, 6 (2011), 362–372.

[8] Bailu Ding, Sudipto Das, Ryan Marcus, Wentao Wu, Surajit Chaudhuri, and Vivek R Narasayya. 2019. Ai meets ai: Leveraging query executions to improve index recommendations. In *Proceedings of the 2019 ACM SIGMOD International Conference on Management of Data*. 1241–1258.

[9] Viktor Leis, Andrey Gubichev, Atanas Mirchev, Peter Boncz, Alfons Kemper, and Thomas Neumann. 2015. How Good Are Query Optimizers, Really? PVLDB 9, 3 (Nov. 2015), 204–215. https://doi.org/10.14778/2850583.2850594

[10] Gabriel Paludo Licks, Julia Colleoni Couto, Priscilla de Fátima Miehe, Renata De Paris, Duncan Dubugras Ruiz, and Felipe Meneguzzi. 2020. SMARTIX: A database indexing agent based on reinforcement learning. Applied Intelligence, 1-14.

[11] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey J. Gordon. 2018. Query-Based Workload Forecasting for Self-Driving Database Management Systems. In *Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data*. 631–645. https://doi.org/10.1145/3183713.3196908

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves et al. 2015. Human-level control through deep reinforcement learning. *Nature*, vol. 7540. 529-533.

[13] Ryan Marcus, and *Papaemmanouil* Olga. *2018. Deep reinforcement learning for join order enumeration. In Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*. 1-4.

[14] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. 2018. The case for learned index structures. In *Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data*. 489–504.

[15] Ankur Sharma, Felix Martin Schuhknecht, and Jens Dittrich. 2018. The case for automatic database administration using deep reinforcement learning .arXiv preprint arXiv:1801.05643.

[16] Karl Schnaitter, Serge Abiteboul, Tova Milo, and Neoklis Polyzotis. On-line index selection for shifting workloads. In *IEEE 23rd International Conference on Data Engineering Workshop*. 459-468.

[17] Richard S. Sutton, and Andrew G. Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[18] Quoc Trung Tran, Ivo Jimenez, Rui Wang, Neoklis Polyzotis and Anastasia Ailamaki. 2015. RITA: An Index-Tuning Advisor for Replicated Databases. In *Proceedings the 27th International Conference on Scientific and Statistical Database Management*, ACM, 1-12.

[19] Transaction Performance Council. TPC-H Benchmark.

[20] CloudLab. https://www.cloudlab.us/.

[21] POWA: PostgreSQL Workload Analyzer (2019). URL http://powa.readthedocs.io

Emotion Cognizance Improves Health Fake News Identification

Anoop K
University of Calicut
Kerala, India
anoopk_dcs@uoc.ac.in

Deepak P
Queen's University Belfast
Northern Ireland, UK
deepaksp@acm.org

Lajish V L
University of Calicut
Kerala, India
lajish@uoc.ac.in

ABSTRACT

Identifying fake news is increasingly being recognized as an important computational task with high potential social impact. Misinformation is routinely injected into almost every domain of news including politics, health, science, business, etc., among which, the fake news in the health domain poses serious risk and harm to health and well-being in modern societies. In this paper, we consider the utility of the affective character of news articles for fake news identification in the health domain and present evidence that emotion cognizant representations are significantly more suited for the task. We outline a simple technique that works by leveraging emotion intensity lexicons to develop emotion-amplified text representations and evaluate the utility of such a representation for identifying fake news relating to health in various supervised and unsupervised scenarios. The consistent and notable empirical gains that we observe over a range of technique types and parameter settings establish the utility of the emotional information in news articles, an often overlooked aspect, for the task of misinformation identification in the health domain.

CCS CONCEPTS

• **Human-centered computing** → *Collaborative and social computing theory, concepts and paradigms*; • **Computing methodologies** → *Natural language processing*; • **Information systems** → *Information systems applications.*

KEYWORDS

Fake News Detection, Health Fake News, Document Emotion

ACM Reference Format:
Anoop K, Deepak P, and Lajish V L. 2020. Emotion Cognizance Improves Health Fake News Identification. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410595

1 INTRODUCTION

The spread of fake news is increasingly being recognized as an enormous problem. In recent times, fake news has been reported to have grave consequences such as causing accidents [15], while fake news around election times has reportedly reached millions of people [1] causing concerns whether they might have influenced the electoral outcome. *Post-Truth* was recognized as the Oxford Dictionary Word of the Year in 2016[1]. These have spawned an extensive interest in the data analytics community in devising techniques to detect fake news in social and online media leveraging content, temporal and structural features (e.g., [13]). A large majority of research efforts on fake news detection has focused on the political domain within microblogging environments (e.g., [9, 15, 16, 23, 32, 33]) where structural (e.g., the user network) and temporal propagation information (e.g., re-tweets in Twitter) is available in plenty.

Fake news within the health domain have been recognized as a problem of immense significance. As a New York Times article suggests, *'Fake news threatens our democracy. Fake medical news threatens our lives'*[2]. This paper is being finalized during the times of COVID-19 when WHO has warned global citizenry against the *'infodemic'*[3], using the term to refer primarily to fake news around the time of the pandemic. Fake health news is markedly different from fake news in politics or event-based contexts on at least two major counts. First, they originate in online websites with limited potential for dense and vivid digital footprints unlike social media channels, and secondly, the core point is conveyed through long, nuanced textual narratives. Perhaps to aid their spread, the core misinformation is often intertwined with trustworthy information. They may also be observed to make use of an abundance of anecdotes, conceivably to appeal to the readers' own experiences or self-conscious emotions (defined in [26]). This makes health fake news detection a challenge more relevant to NLP than other fields of data analytics. In fact, techniques that totally discard content information (e.g., [16, 29]) have met with reasonable success in other domains. Further, a number of fake news sub-categories such as satire, parody, and propaganda are understood to be of much less importance in health fake news (see [28]), making health fake news detection quite a different pursuit at the task level.

We target detection of health fake news within quasi conventional online media sources which contain information in the form of articles, with content generation performed by a limited set of people responsible for it. We observe that the misinformation in these sources is typically of the kind where scientific claims or content from social media are exaggerated or distilled either knowingly or maliciously (perhaps to attract eyeballs). Example headlines and excerpts from health fake news articles we crawled is shown in Table 1. These illustrate, besides other factors, the profusion of trustworthy information within them and the abundantly emotion-oriented narrative they employ. Such sources resemble newspaper

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410595

[1] https://en.oxforddictionaries.com/word-of-the-year/word-of-the-year-2016
[2] https://www.nytimes.com/2018/12/16/opinion/statin-side-effects-cancer.html
[3] https://www.un.org/en/un-coronavirus-communications-team/un-tackling-%E2%80%98infodemic%E2%80%99-misinformation-and-cybercrime-covid-19

Table 1: Examples of health fake news headlines and excerpts from them

Wi-Fi: A Silent Killer That Kills Us Slowly! WiFi is the name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. People can browse the vast area of internet through this wireless device. A common misconception is that the term Wi-Fi is short for "wireless fidelity", however this is not the case. WiFi is simply a trademarked phrase that means IEEE 802.11x. The first thing people should examine is the way a device is connected to the router without cables. Well, wireless devices like cell phones, tablets, and laptops, emit WLAN signals (electromagnetic waves) in order to connect to the router. However, the loop of these signals harms our health in a number of ways. The British Health Agency conducted a study which showed that routers endanger our health and the growth of both, people and plants.
Russian Scientist Captures Soul Leaving Body; Quantifies Chakras It uses a small electrical current that is connected to the fingertips and takes less than a millisecond to send signals from. When these electric charges are pulsed through the body, our bodies naturally respond with a kind of 'electron cloud' made up of light photons. Korotkov also used a type of Kirlian photography to show the exact moment someone's soul left their body at the time of death! He says there is a blue life force you can see leaving the body. He says the navel and the head are the first parts of us to lose their life force and the heart and groin are the last. In other cases, he's noted that the soul of people who have had violent or unexpected deaths can manifest in a state of confusion and their consciousness doesn't actually know that they have died.
Revolutionary juice that can burn stomach fat while sleeping Having excess belly fat poses a serious threat to your health. Fat around the midsection is a strong risk factor for heart disease, type 2 diabetes, and even some types of cancers. Pineapple-celery duo is an ideal choice for those wanting to shed the fat deposits around the stomach area due to the presence of enzymes that stimulate the fat burning hormones. All you need to do is drink this incredible burn-fat sleeping drink and refrain from eating too much sugar and starch foods during the day.

websites in that consumers are passive readers whose consumption of the content happens outside social media platforms. This makes fake news detection a challenging problem in this realm since techniques are primarily left to work with just the article content - as against within social media where structural and temporal data offer ample clues - in order to determine their veracity.

1.1 Our Contribution

In this paper, we consider the utility of the affective character of article content for health fake news detection, a novel direction of inquiry though related to the backdrop of fake news detection approaches that target exploiting satire and stance [6, 24]. We posit that *fake and legitimate health news articles espouse different kinds of affective character that may be effectively utilized to improve fake news detection.* We develop a method to enrich emotion information within documents by leveraging emotion lexicons, which we informally refer as 'emotion amplification'. Our emotion-enrichment method is intentionally of simple design in order to empirically illustrate the generality of the point that emotion cognizance improves health fake news detection within both supervised and unsupervised settings.

While the influence of emotions on persuasion has been discussed in recent studies [18, 27], our work provides the first focused data-driven analysis and quantification of the relationship between emotions and health fake news. Through illustrating that there are significant differences in the emotional character of fake and legitimate news in the health domain in that exaggerating the emotional content aids techniques that would differentiate them, our work sets the stage for further inquiry into identifying the nature of the differences in the emotional content. In short, we devise a methodology to leverage external emotion lexicons to derive emotion-enriched textual documents. Our empirical evaluation depicted in Figure 1 using these emotion-enriched documents for supervised and unsupervised fake news identification tasks establish that emotion cognizance improves the accuracy of fake news identification. This study is orthogonal but complementary to efforts that rely heavily on non-content features (e.g., [29]).

2 RELATED WORK

Our particular task, that of understanding the prevalence of emotions and its utility in detecting fake news in the health domain, has not been subject to much attention from the scholarly community. Herein, we survey two streams of related work very pertinent to our task, that of general fake news detection, and secondly, those relating to the analysis of emotions in fake news.

Owing to the emergence of much recent interest in the task of fake news detection, there have been many publications on this topic in the last few years. A representative and non-comprehensive snapshot of work in the area appears in Table 2. As may be seen therein, most efforts have focused on detecting misinformation within microblogging platforms using the content, network (e.g., user network) and temporal (e.g., re-tweets in Twitter) features in supervised and unsupervised settings [2, 16, 29–31]; some of them, notably [29], target scenarios where the candidate article itself resides outside the microblogging platform, but classification task is largely dependent on information within. An emerging trend, as exemplified by [16, 29], focuses on how information propagates within the microblogging platform, to distinguish between misinformation and legitimate ones. Unsupervised misinformation

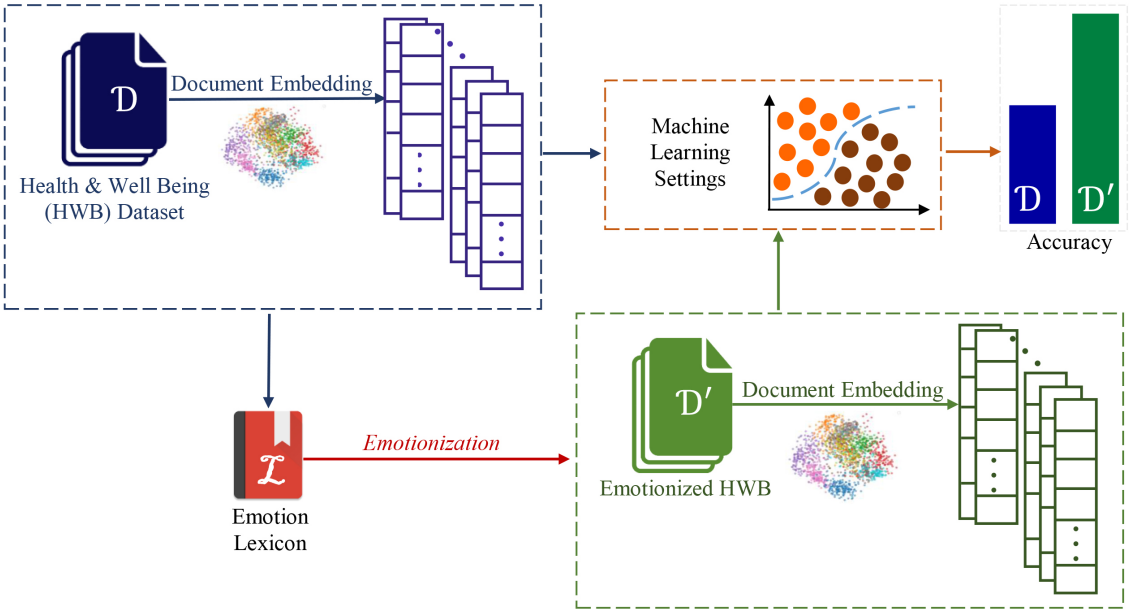


Figure 1: Illustration of the empirical study

detection techniques [30, 31] start with the premise that misinformation is rare and of differing character from the large majority, and use techniques that resemble outlier detection methods in flavor.

2.1 Fake News Detection

Of particular interest are recent works, [5] that make use of sentiment scores and [10] that targets to exploit emotions for fake news detection within microblogging platforms by extensive usage of *publisher emotions* (emotions expressed in the content) and *social emotions* (emotions expressed in the responses) to improve upon the state-of-the-art in fake news detection accuracies. To contrast with these stream of works on fake news detection, it may be noted that our focus is on the health domain where information is usually in the form of long textual narratives, with limited information on the responses, temporal propagation, and author/spreader/reader network structure available for the technique to make a veracity decision.

2.2 Emotions and Fake News

Fake news is generally crafted with the intent to mislead, and thus narratives powered with strong emotion content may be naturally expected within them. The work in [3] analyze fake news vis-a-vis emotions and argue that what is most significant about the contemporary fake news furore is what it portends: the use of personally and emotionally targeted news produced by journalism referring to what they call as “empathic media”. They further go on to suggest that the commercial and political phenomenon of empathically optimised automated fake news is on near-horizon, and is a challenge needing significant attention from the scholarly community.

A recent study, [21], conducts an empirical analysis on 150 real and 150 fake news articles from the political domain and report finding significantly more negative emotions in the titles of the latter. Apart from being distinctly different in terms of domain, our focus being health (vs. politics for them), we also significantly differ from them in the intent of research; our work is focused not on identifying the tell-tale emotional signatures of real vis-a-vis fake news, but on providing empirical evidence that there are differences in emotional content which may be exploited through simple mechanisms such as word-addition-based text transformations. In particular, our focus is on establishing that there are differences, and we keep the identification of the nature of differences outside the scope of our present investigation.

A recent tutorial survey on fake news in social media, [25], places significant emphasis on the importance of emotional information within the context of fake news detection. On a related note, there has been some recent work [22] on using emotional cues within tweets that report exaggerated health news content; it may be noted that the emotion analysis, in this case, is performed on the tweets and not on the articles themselves, making it markedly different from our work in scope.

2.3 Our Work in Context

To put our work in context, we note that the affective character of the content has not been a focus of health fake news detection so far, to our best knowledge. Our effort is orthogonal but complementary to most work described above in that we provide evidence that emotion cognizance in general, and our emotion-enriched data representations in particular, are likely to be of much use in supervised and unsupervised health fake news identification. As observed earlier, identifying the nature of emotional differences between fake

Table 2: Overview of Related Literature

Work	Task Setting	Target Domain	Features Used		
			Content	Network	Temporal
Kwon et al., [13]	Supervised	Twitter	✓	✓	✓
Zubiaga et al., [34]	Supervised	Twitter	✓	✓	✓
Qazvinian et al., [23]	Supervised	Twitter	✓	✓	✓
Wu and Liu, [29]	Supervised	Twitter	✓	✓	✓
Ma et al., [15]	Supervised	Twitter	✓	✗	✓
Zhao et al., [32]	Supervised	Twitter	✓	✗	✓
Ma et al., [16]	Supervised	Twitter	✓	✗	✓
Guo et al., [10]	Supervised	Weibo	✓	✓	✓
Zhang et al., [31]	Unsupervised	Weibo	✓	✗	✓
Zhang et al., [30]	Unsupervised	Weibo	✓	✗	✓

and real news in the health domain is outside the scope of our work, but would evidently lead to interesting follow-on work.

3 EMOTIONIZING TEXT

The intent in this paper is to provide evidence that the affective character of fake news and legitimate articles differ in a way that such differences can be leveraged to improve the task of fake news identification. First, we outline our methodology to leverage an external emotion lexicon to build emotion amplified (i.e., *emotionized*) text representations. The methodology is designed to be very simple to describe and implement, so any gains out of emotionized text derived from the method can be attributed to emotion-enrichment in general and not to some nuances of the method details, as could be the case if the transformation method were to involve sophisticated steps. The empirical analysis of our emotionized representations *vis-a-vis* raw text for fake news identification will be detailed in the next section.

3.1 The Task

The task of emotionizing is to leverage an emotion lexicon $\mathcal{L}$ to transform a text document D to an emotionized document D'. We would like D' also to be similar in format to D in being a sequence of words so that it can be fed into any standard text processing pipeline; retaining the document format in the output, it may be noted, is critical for the uptake of the method. In short:

$$D, \mathcal{L} \xrightarrow{Emotionization} D'$$

Without loss of generality, we expect that the emotion lexicon $\mathcal{L}$ would comprise of many 3-tuples, e.g., $[w, e, s]$, each of which indicate the affinity of a word w to an emotion e, along with the intensity quantified as a score $s \in [0, 1]$. An example entry could be $[unlucky, sadness, 0.7]$ indicating that the word *unlucky* is associated with the *sadness* emotion with an intensity of 0.7.

3.2 Methodology

Inspired by recent methods leveraging lexical neighborhoods to derive word [19] and document [14] embeddings, we design our emotionization methodology as one that alters the neighborhood of highly emotional words in D by adding emotion labels. As illustrated in Algorithm 1, we sift through each word in D in order, outputting that word followed by its associated emotion from the lexicon $\mathcal{L}$ into D', as long as the word emotion association in the lexicon is stronger than a pre-defined threshold τ. In cases where the word is not associated with any emotion with a score greater than τ, no emotion label is output into D'. In summary, D' is an 'enlarged' version of D where every word in D that is strongly associated with an emotion additionally being followed by the emotion label. This ingestion of 'artificial' words is similar in spirit to *sprinkling* topic labels to enhance text classification [11], where appending topic labels to document is the focus. Table 3 shows the emotionized version of the sample article excerpts given in Table 1.

4 EMPIRICAL STUDY

Given our focus on evaluating the effectiveness of emotionized text representations over raw representations, we consider a variety of unsupervised and supervised methods (in lieu of evaluating on a particular state-of-the-art method) in the interest of generality. Data-driven fake news identification, much like any analytics task, uses a corpus of documents to learn a statistical model that is intended to be able to tell apart fake news from legitimate articles. Our empirical evaluation is centered on the following observation: *for the same analytics model learned over different data representations, differences in effectiveness (e.g., classification or clustering accuracy) over the target task can intuitively be attributed to the data representation.* In short, if our emotionized text consistently yields better classification/clustering models over those learned over raw

Algorithm 1: Emotionization

input : Document D, Emotion-Lexicon $\mathcal{L}$, Parameter τ
output : Emotionized Document D'

Let $D = [w_1, w_2, \ldots, w_n]$;
initialize D' to be empty ;
for $(i = 1;\ i \leq n;\ i++)$ **do**
  write w_i as the next word in D' ;
  if $(\exists [w_i, e, s] \in \mathcal{L} \wedge s \geq \tau)$ **then**
    write e as the next word in D' ;
end
output D'

Table 3: Emotionized Health Fake News Excerpts (added emotion labels in bold)

Wi-Fi: A Silent Killer **fear** That Kills **fear** Us Slowly! WiFi is the name of a popular wireless networking technology that uses radio waves to provide wireless high-speed Internet and network connections. People can browse the vast area of internet through this wireless device. A common misconception **fear** is that the term Wi-Fi is short for "wireless fidelity", however this is not the case. WiFi is simply a trademarked phrase that means IEEE 802.11x. The first thing people should examine is the way a device is connected to the router without cables. Well, wireless devices like cell phones, tablets, and laptops, emit WLAN signals (electromagnetic waves) in order to connect to the router. However, the loop of these signals harms **fear** our health in a number of ways. The British Health Agency conducted a study which showed that routers endanger **fear** our health and the growth **joy** of both, people and plants.
Russian Scientist Captures Soul Leaving **sadness** Body; Quantifies Chakras It uses a small electrical current that is connected to the fingertips and takes less than a millisecond to send signals from. When these electric charges are pulsed through the body, our bodies naturally respond with a kind of 'electron cloud' made up of light **joy** photons. Korotkov also used a type of Kirlian photography to show the exact moment someone's soul left their body at the time of death **sadness**! He says there is a blue life force you can see leaving **sadness** the body. He says the navel and the head are the first parts of us to lose **sadness** their life force and the heart and groin are the last. In other cases, he's noted that the soul of people who have had violent **anger** or unexpected deaths **sandess** can manifest in a state of confusion and their consciousness doesn't actually know that they have died **sadness**.
Revolutionary juice that can burn stomach fat while sleeping Having excess belly fat poses a serious threat **anger** to your health. Fat around the midsection is a strong risk **fear** factor for heart disease **fear**, type 2 diabetes, and even some types of cancers **sadness**. Pineapple-celery duo is an ideal choice for those wanting to shed the fat deposits around the stomach area due to the presence of enzymes that stimulate the fat burning hormones. All you need to do is drink this incredible burn-fat sleeping drink and refrain from eating too much sugar and starch foods **joy** during the day.

text, emotion cognizance and amplification may be judged to influence fake news identification positively. This empirical evaluation framework is illustrated in Figure 1. We first describe our Health and Well Being (HWB) dataset, followed by the emotion lexicon used in this work, and then the empirical study settings and their corresponding results.

4.1 Dataset

With most fake news datasets being focused on microblogging websites in the political domain making them less suitable for content-focused misinformation identification tasks as warranted by the domain of health, we curated a new dataset of fake and legitimate news articles within the topic of *health and well being* which being made publicly available at https://dcs.uoc.ac.in/cida/resources/hwb.html. For legitimate news, we crawled 500 health and well-being articles from reputable sources such as CNN, NYTimes, New Indian Express and, many others, manually double-checked for truthfulness. For fake news, we crawled 500 articles on similar topics from well-reported misinformation websites such as BeforeItsNews, Nephef, MadWorldNews, and many others. These were manually verified for misinformation presence as well. Having a good mix of data sources in both fake and real categories, it may be argued, is critical to ensure that the technique is generalizable. The detailed dataset statistics is shown in Table 4.

4.2 Emotion Lexicon

For the lexicon, we use the NRC Intensity Emotion Lexicon [20] which has data in the 3-tuple form outlined earlier. For simplicity, we filter the lexicon to retain only one entry per word, choosing the emotion entry with which the word has the highest intensity. This filtering entails that each word in D can only introduce up to one extra token in D'. To mention concrete statistics, out of 1923 word sense entries that satisfy the threshold $\tau = 0.6$, our filter-out-non-best heuristic filtered out 424 entries (i.e., 22%); thus, only slightly more than one-fifth of entries were affected. This heuristic to filter out all-but-one entry per word was motivated by the need to ensure that document structures be not altered much (by the introduction of too many lexicon words), so assumptions made by the downstream data representation learning procedure such as document well-formedness are not particularly disadvantaged. The emotionization using the filtered corpus was seen to lengthen documents by an average of 2%, a very modest increase in document size. To put it in perspective, only around one in fifty words triggered the lexicon label attachment step, on an average. Interestingly, there was only a slight difference in the lengthening of document across the classes; while fake news documents were seen to be enlarged by 2.2% on average, legitimate news articles recorded an average lengthening by 1.8%. This provides very weak, but initial evidence, that fake news has slightly more emotional content than real ones.

4.3 Supervised Setting

4.3.1 Conventional Classifiers. Let $\mathcal{D} = \{\ldots, D, \ldots\}$ be the corpus of all news articles, and $\mathcal{D}' = \{\ldots, D', \ldots\}$ be the corresponding emotionized corpus. Each document is labeled as either fake or not (0/1). With word/document embeddings gaining increasing

Table 4: Dataset Details

Dataset	Class	Total Number of Documents in the Class	Average Words per Document	Average Sentences per Document	Total Number of Words
Health and Well Being (HWB)	Real	500	724	31	362117
	Fake	500	578	28	289477

popularity, we use the DBOW doc2vec model[4] to build vectors over each of the above corpora separately, yielding two datasets of vectors, correspondingly called $\mathcal{V}$ and $\mathcal{V}'$. While the document embeddings are learnt over the corpora ($\mathcal{D}$ or $\mathcal{D}'$), the output comprises one vector for each document in the corpus that the learning is performed over. The doc2vec model uses an internal parameter d, the dimensionality of the embedding space, i.e., the length of the vectors in $\mathcal{V}$ or $\mathcal{V}'$.

Each of these vector datasets are separately used to train a conventional classifier using train and test splits within them. By conventional classifier, we mean a model such as Naive Bayes (NB), k-Nearest Neighbour (KNN), Support Vector Machine (SVM), Random Forests (RF), Decision Tree (DT) or AdaBoost (AB). The classification model learns to predict a class label (one of *fake* or *real*) given a d-dimensional embedding vector. We use multiple train/test splits for generalizability of results where the chosen dataset (either $\mathcal{V}$ or $\mathcal{V}'$) is partitioned into k random splits (we use $k = 10$); these lead to k separate experiments with k models learnt, each model learnt by excluding one of k splits, and evaluated over their corresponding held-out split. The accuracies obtained by k separate experiments are then simply averaged to obtain a single classification accuracy score for the chosen dataset ($Acc(\mathcal{D})$ and $Acc(\mathcal{D}')$ respectively). Accuracy, a popular measure[5] to evaluate classifiers in binary classification scenarios such as ours, simply measures the sum of true positives and true negatives, and expresses it as a percentage of the dataset size. The quantum of improvement achieved, i.e., $Acc(\mathcal{D}') - Acc(\mathcal{D})$ is illustrative of the improvement brought in by emotion cognizance.

4.3.2 Neural Networks. Neural network models such as LSTMs and CNNs are designed to work with vector sequences, one for each word in the document, rather than a single embedding for the document. This allows them to identify and leverage any existence of sequential patterns or localized patterns respectively, in order to utilize for the classification task. These models, especially LSTMs, have become very popular for building text processing pipelines, making them pertinent for a text data oriented study such as ours.

Adapting from the experimental settings for the conventional classifiers in Section 4.3.1, we learn LSTM and CNN classifiers with learnable word embeddings where each word would have a length of either 100 or 300. Unlike conventional classifiers where the document embeddings are learnt separately and then used in a classifier, this model interleaves training of the classifier and learning of the embeddings, so the word embeddings are also trained, in the process, to benefit the task. The overall evaluation framework remains the same as before, with the classifier-embedding combo being learnt separately for $\mathcal{D}$ and $\mathcal{D}'$, and the quantum by which $Acc(\mathcal{D}')$ surpasses $Acc(\mathcal{D})$ used as an indication of the improvement brought about by the emotionization.

4.3.3 Results and Discussion. Table 5 lists the classification results of the conventional classifiers as well as those based on CNN and LSTM, across two values of d and various values of τ. d is overloaded for convenience in representing results; while it indicates the dimensionality of the document vector for the conventional classifiers, it indicates the dimensionality of the word vectors for the CNN and LSTM classifiers. Classification *models learned over the emotionized text are seen to be consistently more effective for the task*, as exemplified by the higher values achieved by $Acc(\mathcal{D}')$ over $Acc(\mathcal{D})$ (highest values in each row are indicated in bold). While gains are observed across a wide spectrum of values of τ, the gains are seen to peak around $\tau \approx 0.6$. Lower values of τ allow words of low emotion intensity to influence D' while setting it to a very high value would add very few labels to D' (at the extreme, using $\tau = 1.0$ would mean $D = D'$). Thus the observed peakiness is along expected lines, with $\tau \approx 0.6$ achieving a middle ground between the extremes.

The quantum of gains achieved, i.e., $|Acc(\mathcal{D}') - Acc(\mathcal{D})|$, is seen to be notable, sometimes even bringing $Acc(\mathcal{D}')$ very close to the upper bound of 100.0; this establishes that emotionized text is much more suitable for supervised misinformation identification. It is further notable that the highest accuracy is achieved by AdaBoost as against the CNN and LSTM models; this may be due to the lexical distortions brought about addition of emotion labels limiting the emotionization gains in the LSTM and CNN classifiers that attempt to make use of the word sequences explicitly. The best accuracy achieved over $\mathcal{D}'$ at $\tau = 0.6$ is *96.5*, which is better than the best accuracy achieved for $\mathcal{D}$ by 6 percentage points.

4.4 Unsupervised Setting

The corresponding evaluation for the unsupervised setting involves clustering both $\mathcal{V}$ and $\mathcal{V}'$ (Ref. Sec. 4.3) using the same method and profiling the clustering against the labels on the clustering purity measure[6]; as may be obvious, the labels are used only for evaluating the clustering, clustering being an unsupervised learning method. We used K-Means [17] and DBSCAN [7] clustering methods, two very popular clustering methods that come from distinct families. K-Means uses a top-down approach to discover clusters, estimating cluster centroids and memberships at the dataset level, followed by iteratively refining them. DBSCAN, on the other hand, uses a more bottom-up approach, forming clusters and enlarging them by adding proximal data points progressively. Another aspect of difference is that K-Means allows the user to specify the number of

[4] https://radimrehurek.com/gensim/models/doc2vec.html
[5] https://developers.google.com/machine-learning/crash-course/classification/accuracy
[6] https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

Table 5: Classification Results (The numbers are between 0 to 100, and could be interpreted as percentages)

Method	$Acc(\mathcal{D})$	$Acc(\mathcal{D}')$				
		$\tau = 0.0$	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.6$	$\tau = 0.8$
Classification Results for $d = 100$						
NB	77.0	78.0	78.0	78.5	**79.0**	77.5
KNN	75.0	75.0	75.5	76.0	**92.5**	75.0
SVM	50.0	65.0	75.0	75.0	**90.0**	70.0
RF	63.0	71.0	70.0	72.0	**84.0**	80.5
DT	68.0	69.0	70.0	78.0	**94.0**	78.5
AB	55.0	57.0	70.0	71.0	**96.5**	82.5
CNN	87.0	88.0	90.0	88.0	**91.0**	88.0
LSTM	90.5	90.0	91.0	91.0	**92.0**	**92.0**
Classification Results for $d = 300$						
NB	77.0	80.0	81.0	79.0	**83.0**	78.0
KNN	72.0	74.0	75.0	76.0	**91.0**	74.5
SVM	60.0	67.0	72.0	74.0	**89.0**	72.0
RF	65.0	70.0	73.0	71.5	**82.0**	75.0
DT	60.0	65.0	73.0	78.0	**90.5**	75.0
AB	55.0	55.0	72.0	81.0	**94.5**	75.0
CNN	91.2	91.0	**92.7**	92.0	92.0	91.0
LSTM	90.0	90.2	90.0	90.2	**90.7**	90.0

clusters desired in the output, whereas DBSCAN has a substantively different mechanism, based on neighborhood density.

For K-Means we measured purities, averaged over 1000 random initializations, across varying values of k (desired number of output clusters); it may be noted that purity is expected to increase with k with finer clustering granularities leading to better purities (at the extreme, each document in its own cluster would yield a purity of 100.0). For DBSCAN we measured purities across varying values of *ms* (minimum samples to form a cluster); the *ms* parameter is the handle available to the user within the DBSCAN framework to indirectly control the granularity of the clustering (i.e., the number of clusters in the output). Analogous to the $Acc(.)$ measurements in classification, the quantum of purity improvements achieved by the emotionized text, i.e., $Pur(\mathcal{D}') - Pur(\mathcal{D})$, indicate any improved effectiveness of emotionized representations.

We would like to note here that while there are only two labels (*fake* and *real*) that we evaluate clusters against, clusterings which comprise much more than two clusters in the output provide useful evaluation settings. This is because *fake* and *real* articles may appear as various sub-structures in the dataset; these may be intermingled, making it intuitively hard to achieve good accuracies at $k = 2$. In such scenarios where the plurality of underlying clustering structures are expected to map to a small set of labels, a human-in-the-loop process may be naturally envisaged. In this, the human would look at typical documents in each cluster, and assign it one of two labels, and in cases of ambiguous clusters, subject each document in the cluster individually to manual perusal to ascertain the label to be applied. These post-clustering pipelines are significantly advantaged if the clusters are pure (either mostly fake or mostly real), so that manual perusal of individual documents can be avoided. This makes the purity of clusterings that produce much more than two clusters a pertinent measure of interest. Even when there are only two output clusters, manual cluster appraisal and assignment of *fake* and *real* labels is unavoidable since clustering algorithms do not produce labels on their own, being unsupervised methods.

4.4.1 Results and Discussion. Table 6 lists the clustering results in a format similar to that of the classification study. With the unsupervised setting posing a harder task, the quantum of improvements $|Pur(\mathcal{D}') - Pur(\mathcal{D})|$ achieved by emotionization is correspondingly lower. The trends from Table 6 are consistent with the earlier observations in that emotionization has a positive effect, with gains peaking around $\tau \approx 0.6$. The best value achieved with $\mathcal{D}'$ at $\tau = 0.6$ is 88.7%, which is 3.4 percentage points better than the best purity achieved over $\mathcal{D}$. We believe the cause of low accuracy in unsupervised setting is because most conventional combinations of document representation and clustering algorithm are suited to generate topically coherent clusters, and thus fare poorly on a substantially different task of fake news identification.

5 EMOTIONIZATION AND COVID-19 FAKE NEWS

As we finalize this work, many parts of the world are reeling under the COVID-19 pandemic[7]. The core research tasks leading to this work was completed much before COVID-19 erupted. Recently, the direful effects of fake news during the times of COVID-19 pandemic has been called an 'infodemic' by WHO, significantly elevating the relevance of research into combating fake news in the health domain. However, no large-scale datasets of COVID-19 fake news have been made available in the public domain as yet.

[7] https://en.wikipedia.org/wiki/COVID-19_pandemic

Table 6: Clustering Results (The numbers are between 0 to 100, and could be interpreted as percentages)

	$Pur(\mathcal{D})$	$Pur(\mathcal{D}')$				
		$\tau = 0.0$	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.6$	$\tau = 0.8$
k	K-Means Clustering Results for $d = 100$					
2	52.3	52.4	52.3	52.3	**56.1**	52.9
4	78.1	78.0	78.6	79.3	**81.6**	79.3
7	85.0	85.7	85.2	85.1	**86.9**	85.6
10	85.3	85.1	85.1	85.1	**87.7**	85.7
15	85.2	85.3	85.1	85.1	**87.8**	85.8
20	85.2	85.2	85.0	85.1	**88.7**	85.7
k	K-Means Clustering Results for $d = 300$					
2	51.3	52.0	52.0	52.0	**55.5**	52.0
4	77.1	77.8	78.1	78.9	**81.5**	78.5
7	84.0	84.0	85.0	84.9	**86.9**	84.6
10	85.0	85.0	85.0	85.0	**87.1**	85.1
15	85.1	85.3	85.1	85.1	**87.5**	85.2
20	85.0	85.2	85.0	85.0	**88.0**	85.0
ms	DBSCAN Clustering Results for $d = 100$					
20	61.0	62.0	62.0	62.0	**65.0**	61.9
40	62.7	65.5	64.5	58.1	**66.5**	65.0
60	71.6	72.1	72.0	**72.5**	**72.5**	**72.5**
80	85.1	85.0	85.1	85.6	**86.0**	85.6
100	84.5	84.1	84.8	84.7	**86.0**	84.0
ms	DBSCAN Clustering Results for $d = 300$					
20	61.0	61.5	61.0	61.0	**63.5**	62.0
40	63.5	66.3	66.5	66.9	**67.0**	65.5
60	67.5	70.1	70.5	71.0	**71.5**	70.0
80	78.0	81.0	81.9	82.0	**82.5**	80.8
100	75.5	80.0	80.0	80.0	**80.5**	80.0

Among the COVID-19 fake news we have come across, which include fake news on revolutionary juices[8], alcohol bath[9] and cow dung bath[10], we have found significant presence of emotional content in the narratives, indicating the applicability of emotion-oriented fake news detection for identifying COVID-19 fake news. Much of these fake news provide false hope exploiting the widespread fear of the disease and even making targeting the disadvantaged across the economic, political, and socio-cultural spectra[11]. Towards illustrating the emotional content of COVID-19 fake news, we outline the emotionized version of a representative COVID-19 fake news in Table 7. These preliminary qualitative observations indicate that emotion-oriented techniques could be a potential direction for data science research into tackling COVID-19 fake news.

Table 7: An example of Emotionized COVID-19 Fake News (added emotion labels in bold)

Do not consent to nose swab testing! Avoid **fear** the Covid-19 test at all costs. These swabs may be (and probably are) contaminated **fear** with something dangerous **fear**, like viruses or something we don't understand. People should be just as concerned **fear** with the swab as they are about the vaccine. I was wondering why the PCR test for COVID-19 had to be so far back and it got me thinking...how far does it go? So I did some research and found these two pictures and overlapped them. The suprising **joy** evidence was shocking **fear**! The blood **fear** brain barrier **anger** is exactly where the swab test has to be placed.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we considered the utility of the affective character of news articles for the task of fake news detection in the health domain. We illustrated that amplifying the emotions within a news story (and in a sense, uplift their importance) helps downstream

[8] https://thelogicalindian.com/fact-check/lemon-baking-soda-coronavirus-covid-19-kills-20488

[9] https://www.deccanherald.com/national/from-alcohol-bath-to-no-cabbage-here-are-the-covid-19-fake-news-818383.html

[10] https://timesofindia.indiatimes.com/city/dehradun/bathing-in-cow-dung-superstitions-abound-on-how-to-tackle-covid-19/articleshow/74998817.cms

[11] https://www.orfonline.org/expert-speak/how-fake-news-complicating-india-war-against-covid19-66052/

algorithms, supervised and unsupervised, to identify health fake news better. In a way, our results indicate that fake and real news differ in the nature of emotional information within them, so exaggerating the emotional information within both stretch them further apart in any representation, helping to distinguish them from each other. In particular, our simple method to emotionize text using external emotion intensity lexicons were seen to yield text representations that were empirically seen to be much more suited for the task of identifying health fake news.

In the interest of making a broader point establishing the utility of affective information for the task, we empirically evaluated the representations over a wide variety of supervised and unsupervised techniques and methods over varying parameter settings, across which consistent and noteworthy gains were observed. This firmly establishes the utility of emotion information in improving health fake news identification.

6.1 Future Work

Given that our study establishes that there is a notable difference between fake and real news in terms of emotional profiles, we are considering ways of computationally analyzing the nature of the difference in affective character. Further, we are considering developing emotion-aware end-to-end methods for supervised and unsupervised health fake news identification, by blending article emotion cues with collective behavior heuristics that have been effective for fake news identification (e.g., [8]). Secondly, we are considering the use of lexicons learned from data [4] which may be better suited for fake news identification in niche domains. Third, we are exploring the usage of the affective content of responses to social media posts.

ACKNOWLEDGMENTS

The first author was supported by the Rajiv Gandhi National Fellowship (RGNF), University Grants Commission (UGC), India (RGNF-2014-15-SC-KER-79884). The second author was partially supported by Ministry of Human Resource Development, Government of India (MHRD) Scheme for Promotion of Academic and Research Collaboration (SPARC) (Project ID: P620).

REFERENCES

[1] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives* 31, 2 (2017), 211–36.
[2] K Anoop, Manjary P Gangan, P Deepak, and VL Lajish. 2019. Leveraging heterogeneous data for fake news detection. In *Linking and Mining Heterogeneous and Multi-view Data*. Springer, 229–264.
[3] Vian Bakir and Andrew McStay. 2018. Fake news and the economy of emotions: Problems, causes, solutions. *Digital journalism* 6, 2 (2018), 154–175.
[4] Anil Bandhakavi, Nirmalie Wiratunga, P Deepak, and Stewart Massie. 2014. Generating a word-emotion lexicon from# emotional tweets. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)*. 12–21.
[5] Bhavika Bhutani, Neha Rastogi, Priyanshu Sehgal, and Archana Purwar. 2019. Fake news detection using sentiment analysis. In *2019 Twelfth International Conference on Contemporary Computing (IC3)*. IEEE, 1–5.
[6] Sahil Chopra, Saachi Jain, and John Merriman Sholar. 2017. Towards automatic identification of fake news: Headline-article stance detection with LSTM attention models. In *Stanford CS224d Deep Learning for NLP final project*.
[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.
[8] Siva Charan Reddy Gangireddy, Deepak P, Cheng Long, and Tanmoy Chakraborty. 2020. Unsupervised Fake News Detection: A Graph-based Approach. In *31st ACM Conference on Hypertext and Social Media*. ACM, 75–83.
[9] Daniel Gayo-Avello, Panagiotis Takis Metaxas, Eni Mustafaraj, Markus Strohmaier, Harald Schoen, Peter Gloor, Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2013. Predicting information credibility in time-sensitive social media. *Internet Research* (2013).
[10] Chuan Guo, Juan Cao, Xueyao Zhang, Kai Shu, and Miao Yu. 2019. Exploiting emotions for fake news detection on social media. *arXiv preprint arXiv:1903.01728* (2019).
[11] Swapnil Hingmire and Sutanu Chakraborti. 2014. Sprinkling topics for weakly supervised text classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 55–60.
[12] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).
[13] Sejeong Kwon, Meeyoung Cha, Kyomin Jung, Wei Chen, and Yajun Wang. 2013. Prominent features of rumor propagation in online social media. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 1103–1108.
[14] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.
[15] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. (2016).
[16] Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.
[17] James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.
[18] Salman Majeed, Changbao Lu, and Muhammad Usman. 2017. Want to make me emotional? The influence of emotional advertisements on women's consumption behavior. *Frontiers of Business Research in China* 11, 1 (2017), 16.
[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[20] Saif M Mohammad. 2017. Word affect intensities. *arXiv preprint arXiv:1704.08798* (2017).
[21] Jeannette Paschen. 2019. Investigating the emotional appeal of fake news using artificial intelligence and human contributions. *Journal of Product & Brand Management* (2019).
[22] Jasabanta Patro, Sabyasachee Baruah, Vivek Gupta, Monojit Choudhury, Pawan Goyal, and Animesh Mukherjee. 2019. Characterizing the Spread of Exaggerated Health News Content over Social Media. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*. 279–280.
[23] Vahed Qazvinian, Emily Rosengren, Dragomir Radev, and Qiaozhu Mei. 2011. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. 1589–1599.
[24] Victoria L Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the second workshop on computational approaches to deception detection*. 7–17.
[25] Kai Shu and Huan Liu. 2019. Detecting fake news on social media. *Synthesis Lectures on Data Mining and Knowledge Discovery* 11, 3 (2019), 1–129.
[26] Jessica L Tracy and Richard W Robins. 2004. " Putting the Self Into Self-Conscious Emotions: A Theoretical Model". *Psychological Inquiry* 15, 2 (2004), 103–125.
[27] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151.
[28] Przemyslaw M Waszak, Wioleta Kasprzycka-Waszak, and Alicja Kubanek. 2018. The spread of medical fake news in social media–the pilot quantitative study. *Health policy and technology* 7, 2 (2018), 115–118.
[29] Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the eleventh ACM international conference on Web Search and Data Mining*. 637–645.
[30] Yan Zhang, Weiling Chen, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee. 2016. A distance-based outlier detection method for rumor detection exploiting user behaviorial differences. In *2016 International Conference on Data and Software Engineering (ICoDSE)*. IEEE, 1–6.
[31] Yan Zhang, Weiling Chen, Chai Kiat Yeo, Chiew Tong Lau, and Bu Sung Lee. 2017. Detecting rumors on online social networks using multi-layer autoencoder. In *2017 IEEE Technology & Engineering Management Conference (TEMSCON)*. IEEE, 437–441.
[32] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *Proceedings of the 24th international conference on world wide web*. 1395–1405.
[33] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2016. Learning reporting dynamics during breaking news for rumour detection in social media. *arXiv preprint arXiv:1610.07363* (2016).
[34] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting context for rumour detection in social media. In *International Conference on Social Informatics*. Springer, 109–123.

A EMPIRICAL STUDY SETTINGS

We have used the Scikit-learn machine learning library for conventional classifiers and clustering and, Keras neural-network library for CNN and LSTM. In every method, we use default parameters other than some of the important hyperparameters listed below, to aid reproducibility.

A.1 Conventional Classifiers

- NB: *GaussianNB* (Gaussian Naive Bayes algorithm) with default parameters
- KNN: $n\_neighbors = 2$
- SVM: $kernel = linear$
- RF: $max\_depth = 5$, $n\_estimators = 10$
- DT: $max\_depth = 5$
- AB: default parameters

A.2 Neural Networks

We have used the CNN model presented in [12], a neural method that has recorded good performance for text classification, with following hyper-parameters.

- Filter sizes = 3, 4 and 5
- Number of filters = 100
- Embedding dimension, d = 100/300 (Keras Embedding)
- Regularizer = l2(0.01)
- Optimiser = Adam
- Loss = Binary cross-entropy
- Activation function in the dense layer = Sigmoid
- Batch size = 32
- Epoch = 100

The LSTM model is constructed using a single LSTM layer followed by 2 Dense layers, with following hyper-parameters.

- LSTM layer = 100/300 LSTM units
- Dense layer1 = 256 neurons + relu
- Dense layer2 = 1 neuron + sigmoid
- Embedding dimension, d = 100/300
- Optimiser = RMSprop
- Loss = Binary cross-entropy
- Batch size = 32
- Epoch = 100

A.3 Unsupervised Setting

- K-Means: $max\_iter = 500$
- DBSCAN: default parameters

Empowering Big Data Analytics with Polystore and strongly typed functional queries

Annabelle Gillet
LIB Univ. Bourgogne Franche Comté
Dijon, France
annabelle.gillet@depinfo.u-bourgogne.fr

Éric Leclercq
LIB Univ. Bourgogne Franche Comté
Dijon, France
eric.leclercq@u-bourgogne.fr

Marinette Savonnet
LIB Univ. Bourgogne Franche Comté
Dijon, France
marinette.savonnet@u-bourgogne.fr

Nadine Cullot
LIB Univ. Bourgogne Franche Comté
Dijon, France
nadine.cullot@u-bourgogne.fr

ABSTRACT

Polystores are of primary importance to tackle the diversity and the volume of Big Data, as they propose to store data according to specific use cases. Nevertheless, analytics frameworks often lack a uniform interface allowing to fully access and take advantage of the various models offered by the polystore.It also should be ensured that the typing of the algebraic expressions built with data manipulation operators can be checked and that schema can be inferred before starting to execute the operators (type-safe).

Tensors are good candidates for supporting a pivot data model. They are powerful abstract mathematical objects which can embed complex relationships between entities and that are used in major analytics frameworks. However, they are far away from data models, and lack high level operators to manipulate their content, resulting in bad coding habits and less maintainability, and sometimes poor performances.

With TDM (Tensor Data Model), we propose to join the best of both worlds, to take advantage of modeling capabilities of tensors by adding schema and data manipulation operators to them. We developed an implementation in Scala using Spark, providing users with a type-safe and schema inference mechanism that guarantees the technical and functional correctness of composed expressions on tensors at compile time. We show that this extension does not induce overhead and allows to outperform Spark query optimizer using bind join.

CCS CONCEPTS

• **Information systems** → **Query languages**; **Data structures**.

KEYWORDS

High performance data analytics, Polystore, Query language, Tensor

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06.
https://doi.org/10.1145/3410566.3410591

ACM Reference Format:
Annabelle Gillet, Éric Leclercq, Marinette Savonnet, and Nadine Cullot. 2020. Empowering Big Data Analytics with Polystore and strongly typed functional queries. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410591

1 INTRODUCTION AND MOTIVATIONS

The variety and the volume of Big Data have changed the storage needs. Rather than having one storage system to keep everything, a multitude of new specialized storage engines have emerged (e.g. NewSQL, NoSQL, Column Stores, Distributed File Systems, Graph databases), each one corresponding to a use case more or less specific. The well-known article of M. Stonebraker [37], "one size does not fit all", explains that the full potential of data will be better exploited with a polystore architecture. A polystore refers to a system that integrates heterogeneous database engines, storage systems and multiple data manipulation or programming languages using different paradigms [14]. The use of polystore brings several advantages: it allows to organize data according to particular use cases (e.g. graph DBMS well support linked data and graph traversal or path queries); it enables parallel processing among several data store according to the specificities of each kind of system in the polystore [4, 22].

Researches on polystores try to overcome the limitation of traditional tools. Extract Transform Load (ETL) processes and warehousing technologies as well as in-database analysis operators are not sufficient to support complex analytics pipelines. ETL processes are expensive and time consuming tasks, and transforming multiple datasets into a single data model in a data warehouse can impact negatively performances and reduce the expressivity of the original data model. In-database analysis cannot take easily into account new algorithms, as they require a specific development for each database model in order to fit the data structure required by the algorithm [26, 27]. So researches on polystores are directed towards ETL streaming systems [13, 41], multi-database query language [14], unification models [15], and parallel query processing as well as the integration of polystore with analysis frameworks [22]. This article focus on the last three points.

Analytics tasks require multiple kinds of algorithms based on different theoretical foundations, such as linear algebra, statistics,

graph theory. Algorithms are implemented using different computing paradigms such as GPU, map-reduce, concurrent, parallel, or functional programming, and they are used as operators over data in data analytics pipelines. With the adoption of different data models combined with data analysis frameworks and new techniques such as machine learning, we are able to extract more information from data and to have a deeper understanding of the studied phenomena. Nevertheless, data processing pipelines are becoming complex and heterogeneity is no longer observed only at the data level but also at the analysis level [2]. Developing a pivot data model which can generalized polystore underlying data models and which can facilitate data transformations to feed quickly algorithms implemented with frameworks is a major challenge [17, 19, 26].

Multiple data analysis frameworks built on different computing paradigms are available, such as Spark [42] (with SparkSQL, GraphX, MLlib), Tensorflow [1], PyTorch [33], Theano [3], TensorLy [23], NumPy [40]. However, frameworks often focus exclusively on applying algorithms, and not on manipulating or transforming data, even if it is a time-consuming and error-prone task. Each of these tools lacks one or several important properties, such as type-safe functions (operators) which guarantee at compile time that a composition of operations is valid (a property that is automatically lost when using dynamic typing, leading to errors during execution), or schema inference when manipulating data. With the absence of these properties, two categories of errors can arise: 1) type errors, when operators are not applied on the right attribute type and 2) functional errors, when operators are applied on the right attribute type but not on the attribute representing the desired information. The second category of errors is the hardest to detect, but also the most dangerous, because the error will be unnoticed, the computations will occur but will give an incorrect result. For example, an inversion between two columns of a dataset[1] has already led to a major mistake, that has been discovered only years later, and that has resulted in the retraction of five articles and impacted the work of other researchers that were using the erroneous result in their work. This ascertainment highlights the need of checking the correctness of analysis workflows, as shown by the evolution of the data structure in Spark: from *RDD*s with unstructured data, to *DataFrame*s with a columnar format, and finally to *Dataset*s with a type-safe layer over *DataFrame*s. Thus, if composition of operators in programs could have a mean to prevent this type of mistakes, without needing user intervention and without adding complexity to the use, data scientists could focus more on the core of the analysis rather than on controlling the result at different steps of the workflow.

In recent analytics frameworks, tensors play an important role. They are abstract and powerful mathematical objects used in multiple data analytics tools [38], including deep learning to deal with multi-dimensional data or data mining to analyze latent relationships using tensor decompositions [21, 32]. However, their pure mathematical definition is relatively abstract, far away from user needs. Their implementation in tools does not really suit coding standards and brings several bad habits, usually banned from good practices of software development, that reduce evolution, reuse, collaborative work, etc. In frameworks such as Tensorflow, Theano or NumPy, tensors are usually defined as multidimensional arrays; users assign an implicit meaning to dimensions' integer indexes and at best put comments in their code to remember the meaning of constructions [35]. It also may concern dimension names or dimension order before or after tensor transformations are applied. This can lead to errors which cannot be easily understood because the computation is technically correct but functionally incorrect. Moreover, tensors are disconnected from data models, data schemas and data sources, failing to take advantage of the expressiveness of data models and semantically well-defined data manipulation operators. It is necessary to use intermediate data structures to perform complex data manipulations before applying tensorial operators such as decompositions. This is even more true when handling multiple data sources, and strengthens the need of a pivot data model.

In [24] we have formally defined a semantically rich pivot data model, the Tensor Data Model (TDM), by adding schema to the tensor mathematical object. It provides users with operators that can be combined to express complex data transformations. We have showed that tensors make it possible to generalize common data models, and that virtual or materialized views can be defined among multiple data sources (polystore) using operators on tensors. In this article our contribution are the following: we propose a set of mechanisms to ensure type-safe property and schema inference. As we stated, strong static typing is of primary importance in Big Data analytics because it allows to determine errors before the execution and thus to avoid expensive buggy calculation phases which will end with errors or inconsistencies. We describe our implementation of the operators on the top of Spark that fulfill the type-safe and schema inference properties and includes a mechanism to connect to a polystore.

The article is organized as follow: section 2 is a related work on analytics frameworks and their use of tensors, including the role of pivot data model and query processing in polystores; section 3 gives an overview of the key features of TDM; section 4 describes the mechanism for type-safety and schema inference in functional queries; section 5 describes experiments on the top of Spark and shows how to perform optimisations on tensor construct queries.

2 RELATED WORK

This section describes three kinds of inter-related researches: i) support of tensor in analytics frameworks and linear algebra in programming languages ; ii) multidimensional arrays data models in databases, query languages and dataframes ; iii) polystores and their integration in analytics frameworks.

NamedTensor [35] is a first step to make tensors safer and usable in complex workflows. Built on Torch tensor [33], it proposes to use *String* names for dimensions instead of *Integer* indexes. However, Python dynamic typing system does not guarantee safety, and naming dimensions with *String* does not put away the risk of a typo or the referencing to an old dimension that was removed in a code update. In [10], Chen with the Nexus[2] prototype pushes the tensor safety further than NamedTensor, by providing a statically typed tensor abstraction using Scala. Classical tensor operators are defined, but from a mathematical point of view and not from

[1] https://people.ligo-wa.caltech.edu/~michael.landry/calibration/S5/getsignright.pdf

[2] https://github.com/ctongfei/nexus

a data model point of view. So the abstraction level is low. Data transformations are performed using ad-hoc program constructions and not by using well-defined data manipulation operators or a query language.

The works of Muranushi et al. [29] and Griffioen [16] propose a typed linear algebra system, that leads to a more functional and index-free matrix and allows to infer the type of any linear algebra expression. Their implementation in Haskell can detect typing error at compile time. In [29] they apply it for encoding units of measure in astrophysics data.

In the field of data models and query languages, Bauman [7], Libkin et al. [25] have proposed a query language for multidimensional arrays. More recently Brijder et al. [8] study the expressive power of a language for matrix manipulation including linear algebra and graph operations. Barceló et al. [6] study the expressiveness of Lara, a language and a data model built on associative array. The model generalizes standard data models and can also represent tensor but as the building block is an associative array, it forces users to decompose complex relationships into binary ones. Typed arrays are also part of the SQL standard as multi-dimensional arrays [28], but they are mainly designed to be used with an in-database analysis approach or with homogeneous systems, and not with machine learning frameworks and map-reduce paradigm. So, it is much an extension of SQL with new data type rather than a model which is intrinsically based on multi-dimensional structures and can benefit of theoretical results. Spark [42] is a major actor in this field. Its most advanced data structure, the *Dataset*, provides type-safe guarantee at compile time. Unfortunately, schema transformations such as joining two *Dataset*s do not carry on automatically the type-safe property nor automatic schema inference[3].

To integrate different systems in a polystore and to connect it with analytical tools, two approaches can be distinguished: the model approach and the language approach. The goal of the model approach [6, 15] is to build a pivot model that can support all data models of the polystore. However, this approach has often a low level of abstraction, and thus reduces the expressiveness. The language approach [4, 22] defines a multidatabase language to manipulate data. It can be mixed with native queries to directly access a specific storage system. However, operators that need to be applied on a set of results from different data sources have to use the common language, that is often close to the SQL standard. Some works have tried to propose an evaluation framework to measure the ability of a query processing technique over heterogeneous data models [39], that use multiple criteria: the heterogeneity (dealing with several databases without losing expressivity), the autonomy (each store can be managed independently of the polystore system), the transparency (having an easy access to data), the flexibility (building multiple kind of workflows) and optimality (benefiting of the optimization of the stores composing the polystore).

To sum up, tensors are a powerful tool for complex analytics pipelines and for generalizing data models in a polystore architecture. Existing implementations of tensors in analytics frameworks stick to their mathematical nature and lack of operators for manipulating data, though at the center of analysis. Furthermore, the need of a type-safe property for analytics tools is essential, as shown by different works on tensors or on specific data structures. However, type-safety cannot be achieved easily in a dynamically typed language such as Python, and complex data manipulation operators that lead to schema transformation can induce a loss of the type-safe property.

[3]See for example https://medium.com/@pahomov.egor/spark-datasets-are-not-as-type-safe-as-you-think-56a8a9ea0fc

3 AN OVERVIEW OF TDM DATA MODEL

Our aim is to define a pivot model for polystores (figure 1) to benefit from all the different underlying data models, without loosing in expressivity. High level of expressivity is achieved by using tensors, as they have facilities to map to various models, and by allowing native queries for accessing the stores. Moreover, native queries can take advantage of the capabilities of each database individually. The execution of analytics algorithms is then facilitated, as their input data structures are obtained and transformed from TDM and not from each data model underneath. Moreover, the tensorial nature of TDM allows also naturally the use of rich tensorial operators, such as decompositions [34].

3.1 TDM: Algebraic Structure and Operators

Tensors are abstract mathematical objects which can be considered according to various points of view such as family of elements, or multi-linear applications. To be closer to usual definition of data model using set theory we will retain the definition of a tensor as an element of the set of the functions from the product of N sets $I_j, j = 1, \ldots, N$ to $\mathbb{R} : \boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where N is the number of dimensions of the tensor or its order.

For adding the useful notion of schema to tensors we have formally defined the notion of typed associative array and typed tensor [24]. A named and typed associative array is a triple $(Name, \mathbf{A}, T)$ where $Name$ is a unique string that represents the name of a dimension, $\mathbf{A}$ is the associative array (i.e. a map $K \rightarrow \mathbb{N}$ where K is a set of keys), and T is the type of the associative array. The schema of a named typed associative array is $Name : K.\ Dom_{Name}$ is the domain of values taken by the keys of $\mathbf{A}$, i.e., a subset of K. A typed tensor $\boldsymbol{\mathcal{X}}$ is a tuple $(Name, D, V, T)$ where $Name$ is the name of the tensor, D is a list of named typed associative arrays, i.e., one per dimension, V is the values of the tensor and T is the type of the tensor, i.e., the type of its values. The schema of a typed tensor is $Name(S) : T$ where S is the list of schemas of its dimensions, i. e., associative arrays of D. More strictly and by analogy with the relational model, the formal schema of a tensor is the list of names of dimensions to which the name of the tensor is added. For example, the typed tensor $\boldsymbol{\mathcal{UHT}}(User : String, Hashtag : String, Time : Long) : Long$ with the dimensions $User$, $Hashtag$ and $Time$ is used to store the number of times a hashtag is used in tweets produced by a user per time slice.

With its straightforward mapping to diverse data models, TDM is a useful pivot model for polystore architectures. These mappings are presented in detail in [24]. To summarize the main ideas, we focus on the mapping from the most popular data models to TDM:

- **Relational and column**: The mapping from a relation R to typed tensors produces a set of tensors $\boldsymbol{\mathcal{X}}_i$ where the

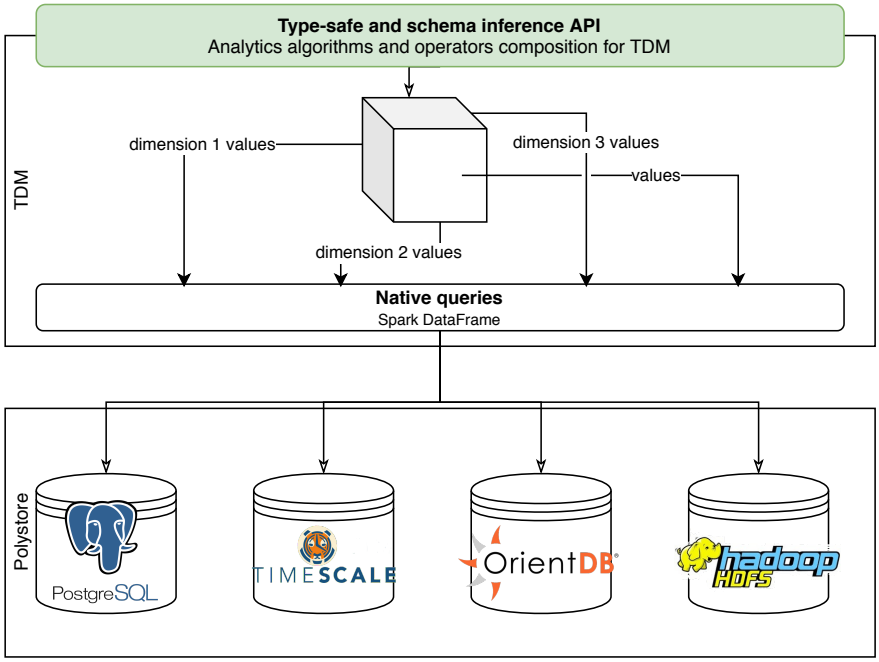


Figure 1: Role of TDM in a polystore system

dimensions D are the n attributes that form together the key of R and for the $k-n$ remaining attributes we create a tensor for each. The keys of each D are formed of different values of each attribute domains.

- **Key-value**: most of key-value stores save data as $(key, value)$ pairs in a distributed hash table. As typed tensors are described by associative arrays, set of $(key, value)$ pairs in NoSQL stores are mapped to a 1-order tensor, with the dimension storing the keys.
- **Graph**: a graph can be represented by an adjacency matrix, i.e. a 2-order tensor representing one of the matrix (e.g. adjacency, Laplacian) which describes the graph.

The definition of typed tensor and tensor schema are the starting point to formally define data manipulation operators on tensors. We have specified projection, selection on the values of tensor, restriction on values of dimension, union, intersection, nesting and natural join. These operators respect the closure property thus allow to define compositions of operators as queries over multiple data sources (i.e. a polystore). By adding difference and Cartesian Product the set of operators is relationally complete [12]. We invite the reader to look at [24] for a formal definition of the operators, that is based on the formalism of [18] used to define the relational model. Our definitions are constructed with two levels:

- a description of the operator's behavior on the schema, i.e., restrictions on operand schemas and specification of the result schema;
- a specification of the operator's semantics on values.

Due to a lack of space we give an outlook of some operators (figure 2 and table 1) in order to illustrate the two levels on which the operators are working on (i.e., the schema level with constraints on the associative arrays values and the value level). The selection, restriction, union and intersection work only at the value level, while the projection, nesting and natural join work at both the value and the schema level.

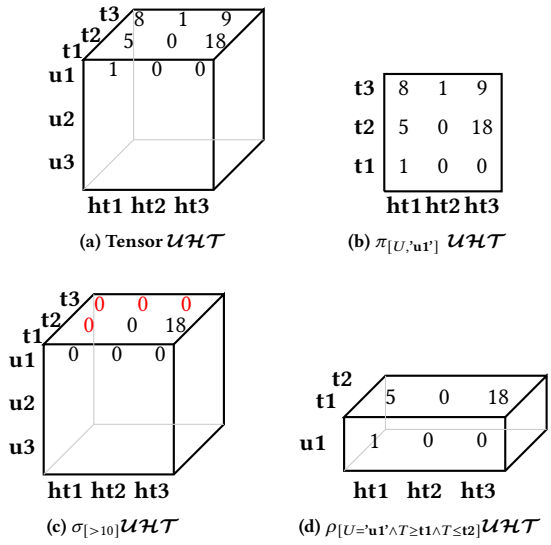


Figure 2: Projection, selection and restriction operators applied to the tensor $\mathcal{UHT}$ for the values of u1 (other values are not shown in the example)

Operator	Signification	Equivalent Relation Algebra Expression	Complexity
$\pi_{[expr]}\boldsymbol{\mathcal{X}}$	projection on some specific value of a dimension, reduce the order by removing the dimension on which the projection is applied	$\pi(\{d_1, ..., d_N\} - d, \boldsymbol{\mathcal{X}})\sigma(d = c)T_{\boldsymbol{\mathcal{X}}}$	$O(1 + \mu_{d,c})$
$\sigma_{[expr]}\boldsymbol{\mathcal{X}}$	selection of values, can reduce the set of values in dimensions	$\sigma(expr)T_{\boldsymbol{\mathcal{X}}}$	$O(1 + nz_{\boldsymbol{\mathcal{X}}})$
$\rho_{[expr]}\boldsymbol{\mathcal{X}}$	reduce a dimension to only values that match the expression, can also reduce the values in other dimensions	$\sigma(expr)T_{\boldsymbol{\mathcal{X}}}$	$O(1 + nz_{\boldsymbol{\mathcal{X}}})$
$\boldsymbol{\mathcal{X}}_1 \cup_\theta \boldsymbol{\mathcal{X}}_2$	union of two tensors having the same schema and perform the θ operation on values having the same keys	$(T_{\boldsymbol{\mathcal{X}}_1} \cup T_{\boldsymbol{\mathcal{X}}_2} - \pi(d_i^{\boldsymbol{\mathcal{X}}_1}, \boldsymbol{\mathcal{X}}_3)T_{\boldsymbol{\mathcal{X}}_3}) \cup \pi(d_i^{\boldsymbol{\mathcal{X}}_1}, \boldsymbol{\mathcal{X}}_1\theta\boldsymbol{\mathcal{X}}_2)T_{\boldsymbol{\mathcal{X}}_3}$ with $T_{\boldsymbol{\mathcal{X}}_3} := \sigma(d_i^{\boldsymbol{\mathcal{X}}_1} = d_i^{\boldsymbol{\mathcal{X}}_2})(T_{\boldsymbol{\mathcal{X}}_1} \times T_{\boldsymbol{\mathcal{X}}_2})$	$O(2 + nz_{\boldsymbol{\mathcal{X}}_1} + nz_{\boldsymbol{\mathcal{X}}_2})$
$\boldsymbol{\mathcal{X}}_1 \cap_\theta \boldsymbol{\mathcal{X}}_2$	intersection of two tensors having the same schema and perform the θ operation on values having the same keys	$\pi(d_i^{\boldsymbol{\mathcal{X}}_1}, \boldsymbol{\mathcal{X}}_1\theta\boldsymbol{\mathcal{X}}_2)T_{\boldsymbol{\mathcal{X}}_3}$	$O(2 + nz_{\boldsymbol{\mathcal{X}}_1} + nz_{\boldsymbol{\mathcal{X}}_2})$
$\boldsymbol{\mathcal{X}}_1 \bowtie \boldsymbol{\mathcal{X}}_2$	join of two tensors having at least one dimension in common and keep the values of the first tensor	$\pi(\bigcup_{\substack{j=1,2 \\ i=1,...N^{\boldsymbol{\mathcal{X}}_j}}} d_i^{\boldsymbol{\mathcal{X}}_j}, \boldsymbol{\mathcal{X}}_1)T_{\boldsymbol{\mathcal{X}}_3}$	$O(\sum_{d \in D^{\boldsymbol{\mathcal{X}}_2}} \lvert Dom_d^{\boldsymbol{\mathcal{X}}_2}\rvert + nz_{\boldsymbol{\mathcal{X}}_1})$

Table 1: TDM operators with their meaning, expression in relational algebra and their complexity using a sparse representation with hashtables for values of dimensions

3.2 Theoretical Complexity

The storage of tensors depends on the nature of data, nevertheless representing and analyzing Big Data with tensor models produces sparsity. For example, in the graph theory, a graph $G = (V, E)$ is considered to be sparse if $|E| = O(|V|)$ resulting in sparse adjacency matrix representation. Compressed Sparse Column (CSC) or Compressed Sparse Row (CSR) are common data structures used to represent sparse matrices [11]. They can be applied directly to tensors, if we consider tensors as set of matrices [9] obtained by unfolding operations. These representations have been extended to sparse tensors with Compressed Sparse Fiber (CSF) format [36]. CSR, CSC and CSF are effective only for some operators such as multiplication and thus are not suitable for supporting the variety of TDM operators.

We will consider two different hypotheses, the first one is the storage of tensors as tuples or elements of a dataframe, the second one, more suitable for in memory storage, is the extension of Knuth structure for sparse matrices [20] (p.302-306) to tensors using hash tables to have direct access to elements sharing the same value on a dimension. The table 1 gives, at the third column, for each operator its cost of execution as a relational algebra expression for the first hypothesis and at the last column its theoretical complexity for the second hypothesis. Notations are the followings: $nz_{\boldsymbol{\mathcal{X}}}$ is the number of existing values in tensor $\boldsymbol{\mathcal{X}}$, $d \in D^{\boldsymbol{\mathcal{X}}}$ is one of the dimension of a tensor, $Dom_d^{\boldsymbol{\mathcal{X}}}$ is the domain, i.e. the set of values for dimension d of a tensor $\boldsymbol{\mathcal{X}}$, $\mu_{d,c}^{\boldsymbol{\mathcal{X}}} = nz_{\boldsymbol{\mathcal{X}}}/|Dom_d^{\boldsymbol{\mathcal{X}}}|$ is an estimation of the number of elements in a sub-tensor when a dimension d is set to a specific value, $T_{\boldsymbol{\mathcal{X}}}$ is a set of tuples in a representation of tensor using relational table or dataframe.

4 STRONGLY TYPED COMPOSITION OF OPERATORS AS A FUNCTIONAL QUERY LANGUAGE

This section presents the mechanisms that allow to leverage the type-safe and schema inference properties. As we put in evidence in sections 1 and 2, these properties are of primary importance for the manipulation and the transformation of data. Once established, they make it possible to build a functional query language based on the composition of TDM operators, possibly including other analysis operators. TDM is developed in Scala, as this language has facilities to establish the mechanisms needed, partially thanks to its strong statically typing system.

4.1 Type-safe and schema inference

In order to have type-safe and schema inference properties in TDM, several mechanisms are needed. In this subsection, we outline phantom types, the shapeless library and implicits, that are the three components of our seeked properties.

Phantom types are types that can never be instantiated. They are used to apply constraints over type, without the need of creating a new object. Their use helps to propagate the type-safe functionality, by allowing the compiler to use these types to check more precise constraints directly over types.

In the TDM library, tensor's dimensions are defined as phantom types, that extends *TensorDimension[T]* with a given type T:

```
object User extends TensorDimension[String]
object Hashtag extends TensorDimension[String]
object Time extends TensorDimension[Long]
```

By doing so, several properties are given to dimensions: 1) each dimension can have a meaningful name, while being of a simple type

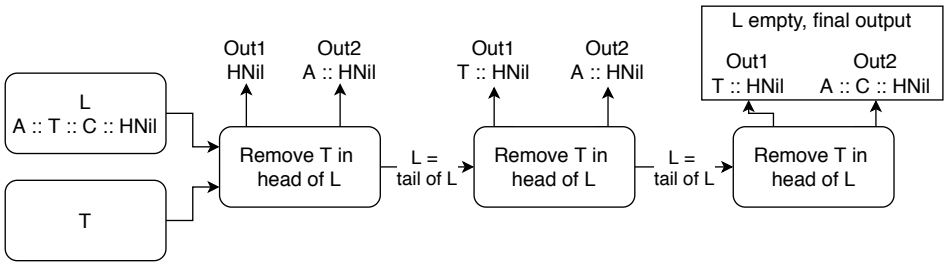


Figure 3: Example: removing all elements of type *T* in a *HList L* with the help of implicits and path dependent type

(such as *String* or *Long*); 2) each dimension is easily identifiable, because it is referenced with a type rather that just a name (e.g. a *String* or an object instance); 3) tensor's dimensions can be controlled more finely, by accepting only once each phantom type but multiple time the same simple type (e.g. for a tensor representing coordinates, two phantom types are used: Longitude and Latitude, each extending *TensorDimension[Double]*); 4) multiple tensors can share a same phantom type, and use it as a constraint to apply operators, thus enforcing the type-safe capability at the schema granularity, with the help of implicits (see below).

With **the shapeless library**[4], some structures to manipulate objects at type level are available. It is the case for the *HList* (Heterogeneous List), that allows to build a list with different types, while keeping the detail of each type and not using a common supertype to work with the list.

In the TDM implementation, a *HList* of phantom types (e.g. *User::Hashtag::Time::HNil*) is used to represent the schema of a tensor, and the type of the tensor is made of a parameterized type. This sort of implementation provides us with tools and methods to interact with the schema of a tensor, that can be triggered with implicits.

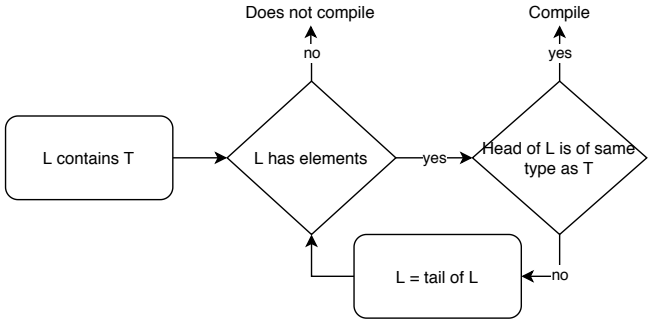


Figure 4: Steps to check if a type *T* is in a *HList L* with the help of implicits

In Scala, **implicits** [30, 31] are a mean to delegate some code logic to the compiler. When a function defines a parameter as implicit, if the user does not provide explicitly the parameter, the compiler will check in the current scope if it is possible to find a value with the corresponding type to use it for the function call.

[4]https://github.com/milessabin/shapeless

This functionality can be pushed to be used to check advanced constraints, or to infer a result type depending on input parameters. If we want to verify that a type *T* is in a *HList L*, we can use implicits in a recursive way (figure 4). We first check if the head of *L* is of the same type as *T*, if it is the case, the implicit compiles because it found a correspondence in its scope. If the head of the list is not of the same type as *T*, the implicit must resolve itself the same implicit, this time called on *T* and on the tail of *L*. If no correspondence is found when reaching the end of *L*, the whole chain of implicits does not compile, thus invalidating the function that used the implicit at the root of the call. By composing these types of constraints, we can build more complex ones, and **use them to enforce validity of operators depending on the schema**.

Implicits can also be used to **infer automatically the schema** of the resulting tensor when applying an operator that alters the schema of the current tensor (e.g. projection, join). For this, implicits are combined with path dependent types [5], that allow to compute a type given one or multiple types in input. The implicit application is the same that for the constraint, but we add the expected output type in the implicit call, and when resolving the implicit, the output type is built. An example of this use is to remove all elements of a given type in a *HList* (figure 3). For an input *HList L*, and a type *T*, we can build two output types: one (*Out1*) as the *HList* containing all the elements of type *T* that were a part of *L*, and one (*Out2*) as the *HList* containing all the elements of *L* except those of type *T*. This removal can be used to apply a projection operator on a tensor, as it removes the dimension on which we want to focus on. The schema of the tensor resulting of the execution of this operator will depend of the schema of the input tensor and of the dimension on which we want to do the projection.

4.2 Towards a functional query language

Scala is a language of choice to develop TDM, because it is statically typed and allows to strongly check type constraints at compile-time (see section 4.1). Our implementation of TDM (available at https://github.com/AnnabelleGillet/TDM) is based on Spark and uses the shapeless library that enables the development of dependent type based generic programs. Above Spark's *DataFrame* we add schema, data manipulation operators and tensor decompositions to implement TDM in a type-safe way. TDM implementation is user-friendly: it hides all the shapeless details from the user and detects errors at compile-time.

A TDM tensor is built in three steps: 1) dimensions are defined; 2) these dimensions are added to the tensor (and can be reused in

other tensors) and 3) values are obtained by querying a data source or manually added. For example, to build the tensor $\mathcal{UHT}$, its three dimensions are created as:

```
object User extends TensorDimension[String]
object Hashtag extends TensorDimension[String]
object Time extends TensorDimension[Long]
```

Notice that dimensions defined in this way, on top of providing properties defined in section 4.1, are also used to help users to produce values for a dimension and to express conditions for data manipulation operators. Dimensions are then used with an object *TensorBuilder*, from which a tensor of type *Long* is instantiated:

```
val tensorUHT = TensorBuilder[Long]
    .addDimension(User)
    .addDimension(Hashtag)
    .addDimension(Time)
    .build()
tensorUHT.addValue(User.value("u1"), Hashtag.value("ht1"),
     Time.value(1))(1)
```

Alternatively, a tensor can also be created by retrieving directly its values from a data source. In this case, users must supply:

(1) a *Properties* object embedding information of connection to the *TensorBuilder*;
(2) the mappings between dimensions and the name of each attribute returned by the query and;
(3) the query to execute and the name of the attribute which contains the tensor's values.

```
val props = new Properties()
...
val query = """SELECT user_screen_name AS user, hashtag,
     published_hour AS time, COUNT(*) AS value
  FROM tweet t INNER JOIN hashtag ht ON t.id = ht.tweet_id
  GROUP BY user_screen_name, hashtag, published_hour HAVING
     COUNT(*) > 5 """
val tensorUHT = TensorBuilder[Long](props)
    .addDimension(User, "user")
    .addDimension(Hashtag, "hashtag")
    .addDimension(Time, "time")
    .build(query, "value")
```

This mechanism allows all storage systems for which a JDBC driver is available to be used as a data source. By using the Spark's data access layer, data sources without a JDBC driver can also be added easily.

TDM operators add data manipulation capabilities to tensors and infer automatically the schema of the resulting tensor, even when the operator induces a schema transformation, e.g. for the natural join. For example, using a projection on the previously defined tensor $\mathcal{UHT}$ for the dimension *User* and the value $u1$ will yield a new tensor $\mathcal{HT}$ populated with the values corresponding to the dimension and the value specified:

```
val tensorHT = tensorUHT.projection(User)("u1")
```

With the naming possibilities of Scala, we can also call the projection operator as:

```
val tensorHT = π(tensorUHT)(User)("u1")
```

Result of this operator can be shown using access to tensor values with the following expressions:

```
tensorHT(Hashtag.value("ht1"), Time.value(1)) // Some(1.0)
tensorHT(Hashtag.value("ht2"), Time.value(2)) // None
```

Other operators can associate two tensors in various ways. For example, the union produces a new tensor with all values from both tensors, and intersection produces a new tensor with only common values between the two tensors. Both operators take a function in parameter that determines the function to apply when keys are in common between tensors.

```
val t1 = tensor1.union(tensor2)((v1,v2) => max(v1,v2))
val t2 = tensor1.intersection(tensor2)((v1,v2) => v1 + v2)
```

The natural join allows to merge schema of two tensors when they have at least one dimension in common. The values kept are those of the first tensor for which the keys combination exists in the second tensor.

```
val t3 = tensor1.naturalJoin(tensor3)
```

TDM operators are implemented with strong type constraints, which can be grouped in three categories: 1) for operators that work at the tensor value level such as selection, the parameter of the condition used must match the tensor value type, 2) for operators working on dimensions that need a dimension parameter, such as the projection or the restriction. The dimension parameter used has to be a part of the schema of the tensor and 3) for binary operators such as union, intersection, natural join or difference. The schema of both tensors must match according to the operators, e.g., for the union and the intersection, the schema of tensors have to be the same and for the natural join the tensors must have at least one dimension in common.

For the internals, TDM uses Spark's *DataFrame*, that correspond to our first hypothesis in section 3.2, and the benefit is multiple: 1) working with a well defined and scalable structure, and 2) keeping the optimization capabilities of Spark. The *DataFrame* is used with n-1 columns for the dimensions' values, and the last column for the value of the tensor associated to these dimensions' values.

The type-safe guarantee is obtained at compile-time by combining shapeless and Scala's implicits, as explained in section 4.1. A compilation-error warns the user if an inconsistency is detected. The use of implicits gives also the capability to define custom compilation-error messages that fit the tensor context, in order to avoid unclear default messages. The following example shows inconsistencies detected at compile time[5]:

```
tensorHT.addValue(Hashtag.value(1), Time.value(2))(2.0) //
     Wrong type of dimension's value
tensorHT.addValue(Hashtag.value("ht2"))(2.0) // Wrong number
     of dimensions
tensorHT.projection(User)("u2") // Dimension not in tensor
tensorHT.union(tensorUHT) // Different schemas of tensors
```

[5]See also https://github.com/AnnabelleGillet/TDM/tree/master/src/test/scala/tdm/core for examples of detected inconsistencies

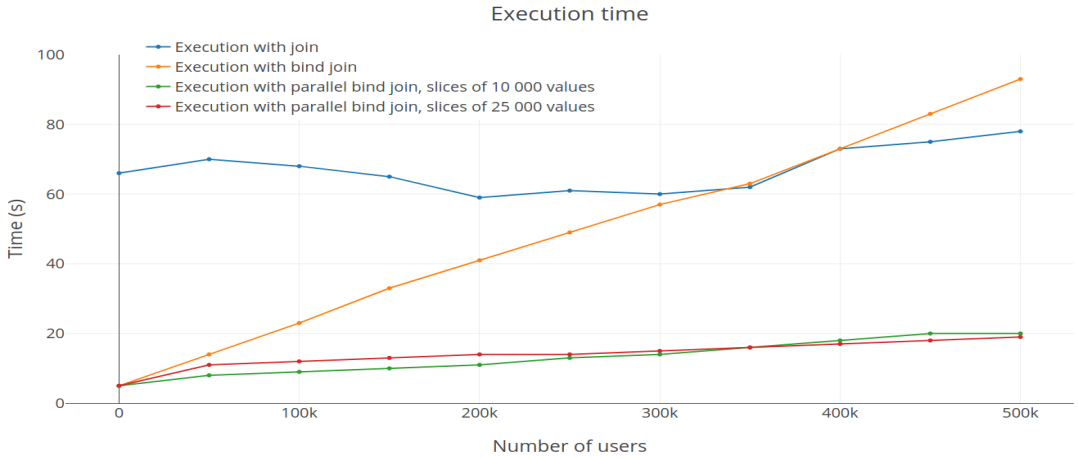


Figure 5: Execution time for a join, a bind join and a parallel bind join, with slices of 10 000 and 25 000 values

Figure 6: Execution time with TDM and Spark

By using the different mechanisms (phantom types and implicits), we showed how the implementation of TDM supports type-safe and schema inference properties. The method used to build tensors by using native queries takes advantage of each database of the polystore, and thus allows to perform fine grained optimizations during the construction with specific techniques such as the bind join.

5 BIND JOIN OPTIMIZATION AND TDM OVERHEAD STUDY: EXPERIMENTS AND RESULTS

In this section, we present two kind of experiments: 1) to show, through a bind join, the benefit of exploiting a polystore and knowing the characteristics of data, and 2) to execute the same combination of operators with TDM and with Spark, in order to see if TDM produces some overhead compared to Spark. The experiments were performed on a Dell PowerEdge R740 server (Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, 20 cores, 256Go RAM).

For the first experiment, we compare a naive join against a bind join that takes advantage of some knowledge about the data. A bind join is an optimisation technique to join a smaller and a bigger sets, where the values of the smaller set are collected and directly sent to the bigger set, in order to minimize data transfers and to limit the bigger set only to values that can actually match with the smaller set. For a set of users $\mathcal{U}$, we want to get $\mathcal{T}$ containing the tweets that a subset of the users have published. These data are stored in a PostgreSQL database, in a table **user** that have 1M elements, and in a table **tweet** that have 50M elements. To see the evolution of the execution time, the subsets of **user** start from 0 to 500 000 elements, with steps of 50 000, and the average execution time of five repetitions is kept.

First, we perform a naive Spark join between $\mathcal{T}$ and each subset of $\mathcal{U}$ (the blue line on figure 5). Then, as we know that $\mathcal{U}$ is smaller than $\mathcal{T}$, we perform a bind join: we only retrieve the values of $\mathcal{T}$ that match those of the subset of $\mathcal{U}$, rather than trying to join all the values. This time, we obtain the orange line in figure 5. The execution time is significantly better than the naive join when

the subset of $\mathcal{U}$ is smaller than 350 000 values. Second, we can push further the exploitation of the knowledge of the data: we know that only one user can have published a tweet, so for two distinct subsets of $\mathcal{U}$, two distinct subsets of $\mathcal{T}$ will be matched. We can split the subset of $\mathcal{U}$ in several slice, run each small bind join against $\mathcal{T}$ in parallel, and then perform an union on all the results and still get the expected result. The green and red lines of figure 5 represent this parallel bind join, with slices of 10 000 and 25 000 values respectively. As we can see, the execution time of our optimization technique is significantly better than the naive and the bind join. At the end of the curve, the bind join with slices of 10 000 values takes a little more time than the one with 25 000 values: it can be explained by the multiplication of the unions that have to be made between each slice, and by the number of queries that have to be sent to the database, when the processor does not have enough cores to handle all the queries at the same time.

With this experiment, we show that a good knowledge of the data and of the behaviour of the database can significantly improve performances compared to a uniform and naive approach.

For the second experiment, we compare an execution with Spark and with TDM. Spark is a well-known and powerful analytics engine, so, an important objective of developing TDM as a layer over Spark is to keep performance equivalent to Spark while taking advantage of the capabilities of TDM. To estimate the eventual overhead induced by TDM, we run an experiment in Spark and compare the result with the same experiment using TDM. The different phases are: 1) building a tensor $\mathcal{U}$ with the users and the number of tweets published by each user as tensor's values, 2) building a tensor $\mathcal{UHT}$ with the number of hashtags published by users for 1h time slices, 3) performing a selection on tensor $\mathcal{U}$ to keep only users who have published at least 100 tweets, 4) joining $\mathcal{U}$ and $\mathcal{UHT}$ to keep the values of $\mathcal{UHT}$ only for active users. We vary the size of tensor $\mathcal{U}$ from 0 to 1M elements by steps of 100 000.

This experiment is carried out in two cases: first by forcing the computation at each operation, and second by forcing the execution only at the last operation, in order to witness the optimization of Spark. The executions are repeated five times, and the average time is measured. A real anonymized data set is available on the github of the experiment [6]. As we can see in figure 6 the execution with TDM does not induce overhead compared to Spark, and the optimization capabilities of Spark are preserved (bottom of fig. 6).

6 CONCLUSION

TDM is a tensor based pivot data model that bridges the gap among data sources and analytics frameworks with a unification of the different theoretical foundations of data models (graph, matrix, relation). The type-safe property and the closure of the operators set are major prerequisites for Big Data analytics. Our library demonstrates that tensors can be manipulated in a safer way, and are well-suited for a data centric use with well-defined data manipulation operators.

The type-safe and schema inference properties of TDM library are implemented by constraints carried out by parameterized types and Scala's implicits. They allow to detect schema inconsistencies and incompatibilities of parameters in operator expressions at compile time. TDM operators allow to build complex expressions over tensors, which can be used as a traditional query language. TDM goes beyond Spark *DataFrame* by providing a layer that keeps the performance of Spark and does not induce overhead, as shown by the comparative experiment between Spark and TDM.

We now focus on developing advanced tensorial operators and algorithms such as hierarchical tensor decomposition, as well as integrating data manipulation operators available for Spark *DataFrame*s that we can use on tensors. We are also studying the capabilities of TDM to take advantage of each database of a polystore depending on the operation optimizations allowed by its model. We plan to develop a mechanism that could optimize the functional queries, by using a context provided by an expert user in order to guide the evaluation depending on the database, as we showed in the particular case of the bind join.

ACKNOWLEDGMENT. This work is supported by ISITE-BFC (ANR-15-IDEX-0003) coordinated by G. Brachotte, CIMEOS Laboratory (EA 4177), University of Burgundy.

[6] https://github.com/AnnabelleGillet/TDM-experiments/tree/master/SparkComparison

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

[2] A. Agrawal, R. Chatterjee, C. Curino, A. Floratou, N. Gowdal, M. Interlandi, A. Jindal, K. Karanasos, S. Krishnan, B. Kroth, et al. Cloudy with high chance of DBMS: A 10-year prediction for Enterprise-Grade ML. In *Conference on Innovative Data Systems Research (CIDR)*, 2020.

[3] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, et al. Theano: A Python framework for fast computation of mathematical expressions. *arXiv:1605.02688*, 2016.

[4] R. Alotaibi, D. Bursztyn, A. Deutsch, I. Manolescu, and S. Zampetakis. Towards Scalable Hybrid Stores: Constraint-Based Rewriting to the Rescue. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1660–1677, 2019.

[5] N. Amin, T. Rompf, and M. Odersky. Foundations of path-dependent types. *ACM SIGPLAN Notices*, 49(10):233–249, 2014.

[6] P. Barceló, N. Higuera, J. Pérez, and B. Subercaseaux. On the Expressiveness of LARA: A Unified Language for Linear and Relational Algebra. *arXiv preprint arXiv:1909.11693*, 2019.

[7] P. Baumann. Management of multidimensional discrete data. *The VLDB Journal*, 3(4):401–444, 1994.

[8] R. Brijder, F. Geerts, J. Van den Bussche, and T. Weerwag. MATLANG: Matrix operations and their expressive power. *ACM SIGMOD Record*, 48(1):60–67, 2019.

[9] A. Buluc and J. R. Gilbert. On the representation and multiplication of hypersparse matrices. In *IEEE International Symposium on Parallel and Distributed Processing*, pages 1–11, 2008.

[10] T. Chen. Typesafe abstractions for tensor operations. In *Proceedings of the 8th ACM SIGPLAN International Symposium on Scala*, pages 45–50. ACM, 2017.

[11] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[12] E. Codd. Relational completeness of data base sublanguages. *Computer*, 1972.

[13] R. C. Fernandez, P. R. Pietzuch, J. Kreps, N. Narkhede, J. Rao, J. Koshy, D. Lin, C. Riccomini, and G. Wang. Liquid: Unifying Nearline and Offline Big Data Integration. In *Conference on Innovative Data System Research (CIDR)*, 2015.

[14] V. Gadepally, P. Chen, J. Duggan, A. Elmore, B. Haynes, J. Kepner, S. Madden, T. Mattson, and M. Stonebraker. The BigDAWG Polystore System and Architecture. In *2016 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2016.

[15] V. Gadepally, J. Kepner, W. Arcand, D. Bestor, B. Bergeron, C. Byun, L. Edwards, M. Hubbell, P. Michaleas, J. Mullen, et al. D4M: Bringing associative arrays to database engines. In *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2015.

[16] P. Griffioen. Type inference for array programming with dimensioned vector spaces. In *Proceedings of the 27th Symposium on the Implementation and Application of Functional Programming Languages*, page 4. ACM, 2015.

[17] H. Jananthan, Z. Zhou, V. Gadepally, D. Hutchison, S. Kim, and J. Kepner. Polystore mathematics of relational algebra. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3180–3189. IEEE, 2017.
[18] P. C. Kanellakis. Elements of relational database theory. In *Formal models and semantics*, pages 1073–1156. Elsevier, 1990.
[19] J. Kepner, V. Gadepally, H. Jananthan, L. Milechin, and S. Samsi. AI Data Wrangling with Associative Arrays. *arXiv preprint arXiv:2001.06731*, 2020.
[20] D. Knuth. *The art of computer programming. Vol. 1: Fundamental algorithms*. Addison-Wesley, 1978.
[21] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
[22] B. Kolev, O. Levchenko, E. Pacitti, P. Valduriez, R. Vilaça, R. Gonçalves, R. Jiménez-Peris, and P. Kranas. Parallel polyglot query processing on heterogeneous cloud data stores with LeanXcale. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1757–1766. IEEE, 2018.
[23] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic. Tensorly: Tensor learning in python. *The Journal of Machine Learning Research*, 20(1):925–930, 2019.
[24] É. Leclercq, A. Gillet, T. Grison, and M. Savonnet. Polystore and Tensor Data Model for Logical Data Independence and Impedance Mismatch in Big Data Analytics. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLII*, pages 51–90. Springer, 2019.
[25] L. Libkin, R. Machlin, and L. Wong. A query language for multidimensional arrays: design, implementation, and optimization techniques. In *ACM SIGMOD Record*, volume 25, pages 228–239. ACM, 1996.
[26] Z. H. Liu, J. Lu, D. Gawlick, H. Helskyaho, G. Pogossiants, and Z. Wu. Multi-model database management systems-a look forward. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, pages 16–29. Springer, 2018.
[27] J. Lu and I. Holubová. Multi-model databases: a new journey to handle the variety of data. *ACM Computing Surveys (CSUR)*, 52(3):1–38, 2019.
[28] D. Mišev and P. Baumann. *SQL Support for Multidimensional Arrays*. IRC-Library, Information Resource Center der Jacobs University Bremen, 2017.
[29] T. Muranushi and R. A. Eisenberg. Experience report: Type-checking polymorphic units for astrophysics research in Haskell. In *ACM SIGPLAN Notices*, volume 49, pages 31–38. ACM, 2014.
[30] M. Odersky, L. Spoon, and B. Venners. *Programming in scala*. Artima Inc, 2008.
[31] B. C. Oliveira, A. Moors, and M. Odersky. Type classes as objects and implicits. *ACM SIGPLAN Notices*, 45(10):341–360, 2010.
[32] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):16, 2017.
[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.
[34] S. Rabanser, O. Shchur, and S. Günnemann. Introduction to tensor decompositions and their applications in machine learning. *arXiv preprint arXiv:1711.10781*, 2017.
[35] A. Rush. Tensor Considered Harmful. Technical report, Harvard NLP, 2010.
[36] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis. Splatt: Efficient and parallel sparse tensor-matrix multiplication. In *IEEE International Parallel and Distributed Processing Symposium*, pages 61–70, 2015.
[37] M. Stonebraker and U. Cetintemel. " one size fits all": an idea whose time has come and gone. In *21st International Conference on Data Engineering (ICDE'05)*, pages 2–11. IEEE, 2005.
[38] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 374–383. ACM, 2006.
[39] R. Tan, R. Chirkova, V. Gadepally, and T. G. Mattson. Enabling query processing across heterogeneous data models: A survey. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3211–3220. IEEE, 2017.
[40] S. Van Der Walt, S. C. Colbert, and G. Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22, 2011.
[41] P. Vassiliadis and A. Simitsis. Near real time ETL. In *New trends in data warehousing and data analysis*, pages 1–31. Springer, 2009.
[42] M. Zaharia and B. Chambers. *Spark: The Definitive Guide*. O'Reilly Media, 2018.

Organizing and Compressing Collections of Files Using Differences

Sudarshan S. Chawathe
chaw@eip10.org
School of Computing and Information Science & Climate Change Institute
University of Maine
Orono, Maine, USA

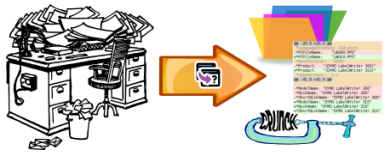


Figure 1: Using a method based on differencing, unorganized document collections may be simultaneously compressed and organized in human-understandable ways.[*]

ABSTRACT

A collection of related files often exhibits strong similarities among its constituents. These similarities, and the dual differences, may be used for both compressing the collection and for organizing it in a manner that reveals human-readable structure and relationships. This paper motivates and studies methods for such organizing and compression of file collections using inter-file differences. It presents an algorithm based on computing a minimum-weight spanning tree of a graph that has vertices corresponding to files and edges with weights corresponding to the size of the difference between the documents of its incident vertices. It describes the design and implementation of a prototype system called *diboc* (for difference-based organization and compression) that uses these methods to enable both compression and graphical organization and interactive exploration of a file collection. It illustrates the benefits of this system by presenting examples of its operation on a widely deployed and publicly available corpus of file collections (collections of *PPD* files used to configure the *CUPS* printing system as packaged by the *Debian GNU/Linux* distribution). In addition to these qualitative measures, some quantitative experimental results of applying the methods to the same corpus are also presented.

CCS CONCEPTS

• **Applied computing** → **Document management and text processing**; • **Information systems** → *Data mining*; • **General and reference** → **Experimentation**; **Empirical studies**; • **Theory of computation** → **Data compression**.

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410584

KEYWORDS

File Collections, Differencing, Compression

ACM Reference Format:
Sudarshan S. Chawathe. 2020. Organizing and Compressing Collections of Files Using Differences. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410584

1 INTRODUCTION

The *main question addressed by this paper* is: How may a collection of related files be processed to yield an equivalent collection that is both compressed and organized in an easily human-understandable manner? The use of the word "related" makes this question a bit imprecise but it is meant to suggest that the documents in collections of interest share at least some significant properties, such as purpose (e.g., system configuration files), document structure, authorship, and language. More precisely, it is reasonable to expect that the efficacy of proposed solutions will depend on the nature and amount of similarity or relatedness of documents in the collection. As well, the interpretation of "easily human-understandable" is necessarily less precise than, for instance, that of compression, for which there are well accepted metrics. Despite these shortcomings, this question is well worth considering due to its practical significance.

An important aspect of this question is the equitable importance of compression and organization. If compression is of overwhelming importance then the very large body of work on compression methods may be profitably used. However, even in such situations, so-called collection-based compression methods have received only a little attention relative to compression methods in general (Section 5). Similarly, if organization is of overwhelming importance then the large body of work on document classification, clustering, and other related information-retrieval methods may be used. In contrast to these two extreme situations, there is much less work

[1]Composite image using clip art from *Openclipart* contributors *Arvin61r58*, *dannya*, *j4p4n*, *jean_victor_balin*, *netalloy*. https://openclipart.org/.

applicable to situations where the two desiderata are more balanced, and these are the situations motivating this work.

The question as posed is still very general and can no doubt be addressed using a variety of techniques. This paper focuses on methods and tools based on differencing, i.e., the automatic computation of differences between documents. In a general sense, differencing (or its dual, similarity-detection) forms the basis of many of the popular compression schemes. However, such uses of differencing tend to be at too low a level of abstraction for easy human comprehension. For instance, the task of determining how a later part of a document differs from an earlier part based on compression dictionaries as used by LZW and related schemes is not at all appealing. When such low-level differences refer to not only one document but potentially several in a large collection, the task is even less inviting and practically infeasible. This work therefore focuses on the use of *human-understandable differences*.

Some *contributions* of this paper may be summarized as follows:

- It motivates the need for methods that can simultaneously compress and organize a collection of related documents in a human-understandable manner.
- It proposes methods based on computing and organizing inter-document differences. This choice is motivated in large part by the wide availability and popularity of differencing tools as commonly used by practitioners in version-control repositories (e.g., GitHub, GitLab) and other systems. This popularity of differencing tools addresses the human-understandable aspect of the primary question.
- It formalizes this problem as that of computing a minimum (cost) spanning tree of a graph defined on a collection of documents, with vertices representing documents and weighted edges representing the sizes of the differences between documents.
- It outlines the design and implementation of a prototype called *diboc* (*diff*-based organization and compression) based on these ideas.
- It describes the results of experimentally evaluating the method (and implementation) on a well known publicly available corpus of document-collections. Given the nature of the question, this evaluation is both qualitative (cf. human-understandable organization) and quantitative (cf. compression).

Some of the paper's *results* may be summarized as follows:

- On at least one practically significant collection of documents in widespread use, differencing-based simultaneous organization and compression works very well, providing competitive compression ratios with the significant added benefit of human-understandable organization.
- Initial experience with the *diboc* prototype provides encouraging anecdotal evidence of its benefits beyond compression. In particular, the ability to view a well-rendered minimum-spanning tree of the document-differences graph, and to interactively examine the documents and differences is very valuable in understanding a document collection.
- Extracting a document from the compressed representation as implemented in *diboc* can be done in time that compares favorably with standard tools despite the latter using low-level representations.
- The time for building the compressed representation is dominated by the time for computing differences. This time grows quadratically with collection size and is thus of limited scalability. In applications such as software-system packaging, where the compression is infrequent and typically done on a well provisioned computer, in contrast to decompression that is performed very frequently in diverse environments, this drawback is not too severe. Nevertheless, a promising avenue for continuing work is the use of approximate or probabilistic methods for this purpose.

Paper outline: Section 2 elaborates on the above file-collection organization and compression problem in the context of a concrete corpus of file collections that is of practical significance. A specific differencing-based variant of this general problem is developed in Section 3. That section outlines some of the important options, explains the specific choices made in this work, and maps the resulting specific variant to the well known problem of computing a minimum cost spanning tree of a weighted graph. It also includes a very small, but real and illustrative example of the method and implementation in action. Section 4 includes brief notes on the implementation of the *diboc* prototype before a summary of experimental results on the organization and compression aspects. Related work is addressed in Section 5. Section 6 summarizes the work and outlines ongoing efforts.

2 COLLECTIONS OF RELATED DOCUMENTS

The initial impetus for this work was provided by a chance examination of some packages used by the printing subsystem of *Debian GNU/Linux* (as well as many other POSIX-like systems). Specifically, the *CUPS* printing system and others like it rely on collections of so-called *PPD* (originally PostScript Printer Description) files [13]. A representative excerpt appears in Fig. 2. These collections are typically organized by hardware vendors or collection curators and some of them are summarized by Table 1. The collections range from tiny (including one with just a single PPD file) to quite large (several thousand PPD files) and the largest has an uncompressed size over 350 MB. While printers are notorious for quirks and special-handling requirements that may explain some of this mass of information, it is nevertheless unclear how similar or diverse the individual PPDs are. Given the favorable compression ratios, it is tempting to assume a high degree of similarity. However, compression algorithms are very highly developed and it is not clear that such similarity manifests itself in a manner that is easily comprehensible to humans. We thus arrive at the main question: How may we process these collections to make the document collections easier to understand en masse? Further, could such processing also permit competitive compression of the collections?

While the specific corpus summarized by Table 1 is valuable because it provides a concrete use case and basis for experimental evaluation, the question as posed above is applicable in a much wider context, and not at all specific to printing systems or even system configurations. Rather, it may be understood in the general context outlined in Section 1, and is that of simultaneously aiding the human comprehension of collections of related documents and

Table 1: Summary of a motivating corpus of document-collections: PPD (PostScript Printer Description) files used by the CUPS printing system. The last four columns report on the use the popular XZ compression scheme [8] in two ways: The *xz* columns report the size and compression ratios when XZ is applied to the concatenation of all documents. The *xz1* provide similar information when the documents are first individually compressed and then concatenated.

id	document collection	files	unique	index size	total size (bytes)	xz size	xz1 size	xz cr	xz1 cr
dym	dymo	25	25	4 130	951 050	21 880	154 404	0.023	0.162
foo	foo2zjs	91	91	23 476	2 343 031	31 472	380 080	0.013	0.161
pto	ptouch	22	22	5 613	391 443	12 652	78 992	0.032	0.199
esc	escpr	505	505	171 774	25 607 270	198 264	2 974 184	0.008	0.115
fom	foomatic-db-compressed-ppds	4 108	4 108	1 139 793	57 750 807	378 428	13 067 172	0.006	0.222
fuj	fujixerox	1	1	205	24 814	9 948	4 120	0.398	0.165
m23	m2300w	4	4	1 028	54 792	8 896	12 152	0.159	0.218
pxl	pxljr	3	3	637	56 748	9 820	11 632	0.171	0.203
ope	openprinting-ppds	4 010	3 827	903 562	378 404 809	4 591 724		0.012	
pos	postscript-hp	606	282	124 069	42 485 588	1 018 988		0.024	

of compressing them. The comprehension aspects are addressed in this work only anecdotally, but using real data and examples that suggest merit. The compression aspects are addressed using the usual metrics of compression ratio, compression time, and decompression time. The time required to extract a single document from a compressed collection is of particular interest, in part because of its importance in applications of the PPD corpus, which typically need only single-digit numbers of documents from collections containing potentially thousands.

3 DIFFERENCING-BASED ORGANIZATION AND COMPRESSION

Simply browsing the documents in collections of even very modest sizes is not very effective in discovering inter-document relationships, and certainly does not scale beyond collections of a few dozen documents at most. This difficulty of determining the differences between documents that are likely to be very similar is not new and one that has led to several algorithms and tools for automatic differencing, perhaps the most notable being the *GNU diff* [16] family. An small example of such a *diff* appears in Fig. 3. These methods and tools operate on pairs, and sometimes triples, of documents and not on large collections at once; therefore they do not immediately address the main question here. Nevertheless, they suggest pairwise document differences as a potential basis for further organization and compression of a collection. Work on collection-based compression (e.g., VCDIFF [14], FemtoZip [20]), of either a static or streaming variety, is related but focuses on only the compression aspect of the question and not the organization and understanding aspects.

Even with the main question narrowed to focus on differencing-based methods, there is a plethora of choices for options such as the document model (strings, bag of words, structured, etc.), differencing algorithm, algorithmic options, and so on. For concreteness, the presentation focuses on *GNU diff* with the commonly used *unified diff* option. The latter may seem a curious choice if one focuses on the compression aspect because it is not as compact as some of the other choices. Its choice is motivated by its widespread use in practice (as a de facto standard) and the ensuing ease and frequency with which a large population of programmers and other practitioners use it. Nevertheless, much of what ensues is also applicable to other diff formats and algorithms.

Once the differencing algorithm and options are sufficiently narrowed as above, an *edit distance* between two documents may be unambiguously defined. As with differencing and diff formats, there are several viable options for the details of such a definition. For reasons similar to those motivating the choice of the diff format above, the sequel uses a simple definition of this distance: It is the length (in bytes) of the unified diff of the two documents. The evaluation considers both uncompressed and compressed sizes of the unified diff (separately) as options for this distance. Compressed diffs are clearly incomprehensible to humans but if they are individually compressed (or compressed using a suitable collection-compression scheme) then they may be decompressed rapidly as needed. The possibility that some diffs may be more compressible than others (by the selected compression scheme at least) suggests exploring both the compressed and uncompressed diffs.

With all the above options fixed for the present, the remainder of the main problem may now be mapped to the well studied minimum (cost) spanning tree problem as follows: Consider a document collection C and an edit-distance function $d : C \times C \to \mathbb{R}$ that maps pairs of documents to a numeric representation of the difference between them (edit distance, length of the edit script, or size of the unified diff). The problem is then that of computing a minimum spanning tree of the *diff-distances graph*, defined as the weighted (undirected) graph $G(C, d) = (V, E)$ with vertices $V = C \cup \{\perp\}$ and edges $E = V \times V$ and edge weights given by $w(u, v) = d(u, v)$ if $u, v \neq \perp$ and $w(\perp, v) = |v|$ where we use $|v|$ to denote the size (in bytes) of the document (vertex) v. The special *initial* vertex $\perp$ is added to ensure both that at least one document in the collection is stored in its entirety (to enable other documents to be derived from it using suitable diffs) and that a diff is not used if it would be more economical to store the entire document instead.

Fig. 4 illustrates these ideas using a very small (but real) collection of just four PPD files. It is the unedited output produced

```
[...]
*Manufacturer:  "DYMO"
*Product:   "(DYMO LabelWriter 315)"
*ModelName: "DYMO LabelWriter 315"
*NickName: "DYMO LabelWriter 315"
*ShortNickName: "DYMO LabelWriter 315"
*APPrinterIconPath: "/Library/Printers/DYMO/CUPS/Resources/
      LW315.icns"

*cupsIPPReason com.dymo.out-of-paper-error/Out of labels. : ""
*cupsIPPReason com.dymo.read-error/Cannot read data from
      printer. : ""
[...]
*OpenUI *PageSize/Media Size: PickOne
*OrderDependency: 10 AnySetup *PageSize
*DefaultPageSize: w79h252
*PageSize w72h154/11352 Return Address Int: "<</PageSize[72
      154]/ImagingBBox null/cupsInteger0 0>>setpagedevice"
*PageSize w72h72/11353 Multi-Purpose: "<</PageSize[72 72]/
      ImagingBBox null/cupsInteger0 0>>setpagedevice"
*PageSize w54h144/11355 Multi-Purpose: "<</PageSize[54 144]/
      ImagingBBox null/cupsInteger0 0>>setpagedevice"
*PageSize w79h252/30252 Address: "<</PageSize[79 252]/
      ImagingBBox null/cupsInteger0 0>>setpagedevice"
[...]
```

Figure 2: An excerpt of a sample PPD file from the corpus of Table 1.

```
--- lw300.ppd 2020-06-02 11:31:13.105756172 -0400
+++ lw310.ppd 2020-06-02 11:29:35.458069211 -0400
@@ -2,3 +2,3 @@

-*% $Id: lw300.ppd 16401 2011-10-31 18:51:16Z pineichen $
+*% $Id: lw310.ppd 16401 2011-10-31 18:51:16Z pineichen $

@@ -25,5 +25,5 @@
 *LanguageEncoding: ISOLatin1
-*PCFileName:    "LW300.PPD"
+*PCFileName:    "LW310.PPD"
 *Manufacturer:  "DYMO"
-*Product:   "(DYMO LabelWriter 300)"
+*Product:   "(DYMO LabelWriter 310)"
 *cupsVersion:   1.2
@@ -41,5 +41,5 @@

-*ModelName: "DYMO LabelWriter 300"
-*NickName: "DYMO LabelWriter 300"
-*ShortNickName: "DYMO LabelWriter 300"
+*ModelName: "DYMO LabelWriter 310"
+*NickName: "DYMO LabelWriter 310"
+*ShortNickName: "DYMO LabelWriter 310"
 *APPrinterIconPath: "/Library/Printers/DYMO/CUPS/Resources/LW310.icns"
@@ -488,3 +488,3 @@
 *%
-*%  End of "$Id: lw300.ppd 16401 2011-10-31 18:51:16Z pineichen $"
+*%  End of "$Id: lw310.ppd 16401 2011-10-31 18:51:16Z pineichen $"
 *%
```

Figure 3: Sample output of *GNU diff* [16], in the *unified diff* format, on two files from the corpus of Table 1.

in a fully automated manner by the *diboc* prototype on the m23 collection of Table 1. The edge labels in the figure are the sizes (in bytes) of the unified diffs (uncompressed) of the PPD files identified by the incident vertices. Edges incident on the i ($\perp$) vertex are labeled with the size in bytes of the document identified by the other vertex. Although this document collection is almost comically small, the *diboc* output is still very useful in determining its structure and inter-document relationships. A closer examination of the documents (as facilitated by *diboc*) reveals that the only real difference between PPDs with identifiers 0 and 2 (and likewise 1 and 3) is the manufacturer name. The difference between the other pairs is less trivial but still very small. It is worth emphasizing that even for this tiny collection, discovering these relationships by simply visually examining the documents is not easy. Since $\binom{4}{2} = 6$

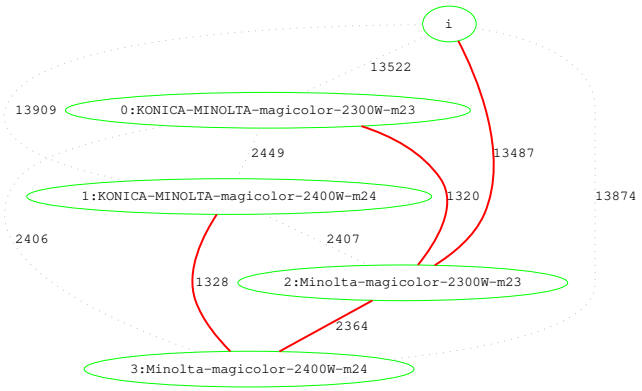


Figure 4: The diff-distances graph and a minimum spanning tree (MST) for the very small m23 dataset (m2300w, 4 files). The node labeled "i" represents an initial, empty file. The other node labels identify files in the collection using the format *file-index*:*short-name*. Labels on edges are the sizes (in bytes) of the *unified diff* of the files represented by the corresponding vertices. Bold, red edges belong to the MST.

is a very manageable number, one may potentially compute the 6 diffs and their visual examination is likely to provide similar clues. However, that task is tedious at this tiny scale and very quickly becomes infeasible for larger collections. An automated method is therefore essential. Finally, the weights on the graph edges indicate that a substantial savings in storage may be realized by storing the diffs instead of all the base documents, even if the diffs are stored uncompressed.

4 IMPLEMENTATION AND EXPERIMENTAL EVALUATION

The implementation of the *diboc* prototype by design draws on several well established implementations and techniques. One of the motivations for the implementation is to suggest an alternative encoding for the PPD document collections of Table 1 for inclusion in software distributions. In this context, the use of a well established code-base is a significant advantage. The experimental prototype is implemented on the Java Virtual Machine (JVM) platform using the *Kawa Scheme* language [3]. Differencing is implemented using the *java-diff-utils* library [19]. XZ compression and decompression is implemented using *XZ Utils* [8]. Minimum spanning trees are implemented with *JGraphT* [17]. Layout of the diff-distances graphs and minimum spanning trees uses *GraphViz* [11] An important aspect of the implementation is the use of well documented and standardized APIs, facilitating porting to other environments, such as a Python implementation potentially more suitable for software packaging and distribution.

As is often the case when working with long-lived databases with information from disparate sources, the PPD collections in the corpus of Table 1 require careful handling in the implementation. For example, although PPD files are text documents, and although the vast majority appear to use the Unicode UTF-8 encoding, that is not true of all the documents. Therefore, it is necessary to read

and process the files as binary data in order to preserve document integrity. (A similar observation appears as a comment in *pyppd* [13] source code.)

Another pragmatic concern, although a bit tangential to the primary focus of this work, is that of identifying the files in a collection in a manner that is intuitive to humans. It is quite common for files in large collections to have long and unweildy names, such as `cups-model-epson-inkjet-printer-escpr-Epson-Artisan-710-epson-escpr-en.ppd` in the corpus of Table 1. While such identifiers pose no challenges in machine use, they are very unsuitable in depictions for human use. For instance, the visual clutter and layout difficulties resulting from the use of such unweildy identifiers in depictions such as Figs. 5 and 6 is easy to visualize. (Indeed, even a textual presentation of this identifier in the present document requires special handling for hyphenation in order to avoid exceeding the printable areas.) The *diboc* implementation includes a simple but effective scheme for unambiguously abbbreviating names in file collections for this purpose, and yields short yet meaningful identifers of the kind appearing in Figs. 5 and 6.

4.1 Organization

Figs. 5 and 6 illustrate the output of the *diboc* system on the `dym` and `pto` collections of Table 1. Although both collections are of very modest size, they are large enough to make any kind of exhaustive pairwise computation and examination of differences by humans impracticable, with $\binom{25}{2} = 300$ pairs for `dym` for instance. The figures use the same conventions as Fig. 5 which was described earlier, except that non-MST edges are elided for clarity. As before, the figures represent the unedited output of *diboc* operating on the document collections with no human guidance. The only exception is the inset in Fig. 5, which was edited in for compact presentation. Examining the MST reveals in only a few seconds the structure (more precisely, a plausible structure) within the collections. Examining a few inter-document differences, as suggested by the inset, which presents the complete diff between the selected documents, reveals additional useful information, such as these two PPDs differing only in the identifiers used for the models and not in any substantial system details. It is worth noting that the MST allows a human to significantly narrow the choices for inter-file differences that may be worth examining visually and the *diboc* implementation both provides the MST in an easily comprehensible format and facilitates examination of diffs. The significance of this facility is very apparent when one is confronted with a file-system directory (folder) containing even this modest number of files: How does one proceed otherwise to attempt to gain an understanding of the files in the collection?

Fig. 7 presents similar results on the `foo` collection of Table 1. As before, the figure presents the verbatim result of the unaided *diboc* implementation on this collection, with two caveats: (1) The insets are edited in for presentation reasons. (2) Some abbreviations were manually suggested to *diboc* in order to allow compact presentation (e.g., HPLJ for HP-LaserJet, and initials for other manufacturers). (Most other abbreviations from the typically very long names were generated unaided, as in earlier figures.) Examining this MST for a few seconds reveals several useful relationships among the documents and the facilitated visual examination of the differences allows elaboration. As an example, we may note that the early model HPLJ-1000 is chosen as the primary document for this collection (connected to the *init* node) which is intuitively reasonable. (The *diboc* implementation does not have any special coding for model names and numbers and this structure is determined solely based on the diff-distances graph.) It may seem curious that the HPLJ-1018 vertex is linked to HPLJ-1022 instead of HPLJ-1020 which one may reasonably expect to be closer (although model numbers certainly do not exhibit such consistency in general). Examining the edge weights, along with the two pairwise diffs indicated by the color-filled vertices and the two insets provides an explanation similar to that for the example of Fig. 4: There are only relatively trivial naming and device-id differences differences.

Fig. 8 presents the slightly edited result of *diboc* on the much larger `esc` collection of Table 1. The only edit is the elision of all labels other than the integer identifiers of vertices (which may be correlated to the underlying files using a manifest). The sole purpose of this edit is to facilitate presenting the result in a non-interactive paper format here. It is reasonably easy to browse the unedited output, which includes labels, interactively by panning and zooming the window. The main observation here is that, even with labels elided, this organization of the files provides a human with significant guidance on which documents and inter-document differences may be worth examining. For example, the documents with identifiers 114 and 260 appear to be significant. (The vertex with identifier 506 is the *init* vertex.)

4.2 Compression

In interpreting the compression ratios summarized by Fig. 9, it is important to recall that the *xz-m* scheme simply compresses the concatenation of all files in the collection using the opaque (unintelligible to humans) XZ compression scheme. Thus it is not surprising that it provides a very good compression ratio but it is most appropriate to regard it as a practical lower bound rather than as a real option for the problem addressed by this work. In this regard, the performance of the `xdt` scheme is quite attractive, when one considers that it permits the kinds of organization and exploration described earlier, which is not possible with the compression-only *xz-m* scheme.

The single-file extraction times (averaged over all files of the collection) for the four methods are summarized by Fig. 10. These results clearly illustrate the significant weakness of the *xz-m* method even when focusing solely on the compression aspect of the problem (since it is purely a compression method that does not provide any guidance on organization of collections). The performance of the *xz-1* scheme, which compresses files in the collection individually is significantly better, as may be expected. In this context, an interesting observation is that the performance of the `dt` scheme, which stores uncompressed diffs, is very competitive and appealing for its combination of performance and simplicity (albeit at the expense of a poorer compression ratio as summarized by Fig. 9).

The time required for the complete processing of file collections, including organization and compression as applicable, for the four methods is summarized by Fig. 11, with a logarithmic scale on the vertical axis. Here, the weakness in scalability of the difference-based *dt* and *xdt* methods is apparent, especially for the larger `foo`

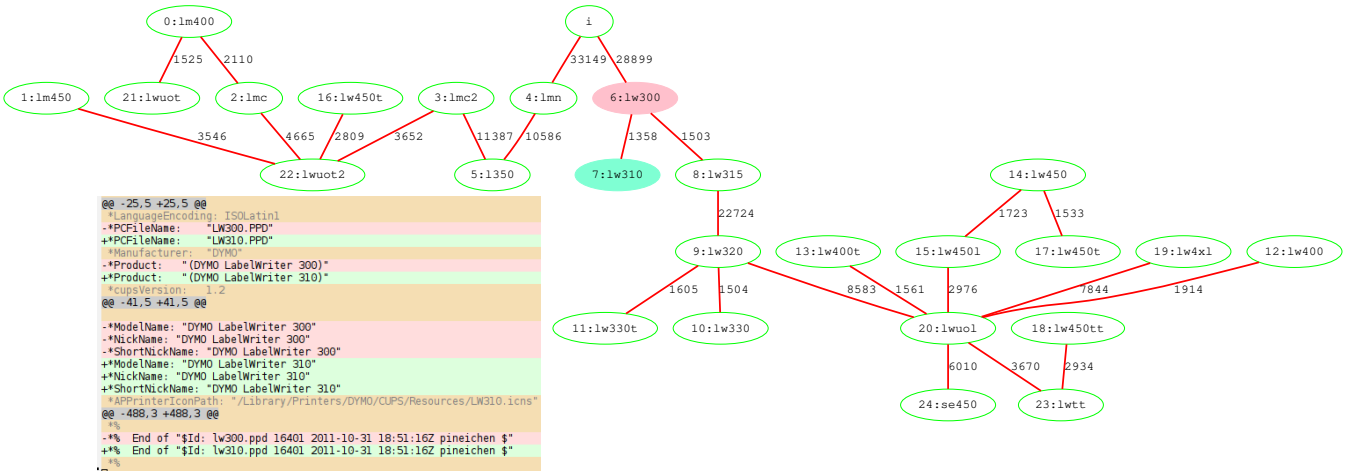


Figure 5: The minimum spanning tree (MST) of the diff-distances graph for the dym dataset (dymo, 25 files). The details are similar to those for Fig. 4 but, in this figure, non-MST edges of the diff-distances graph are omitted for clarity.

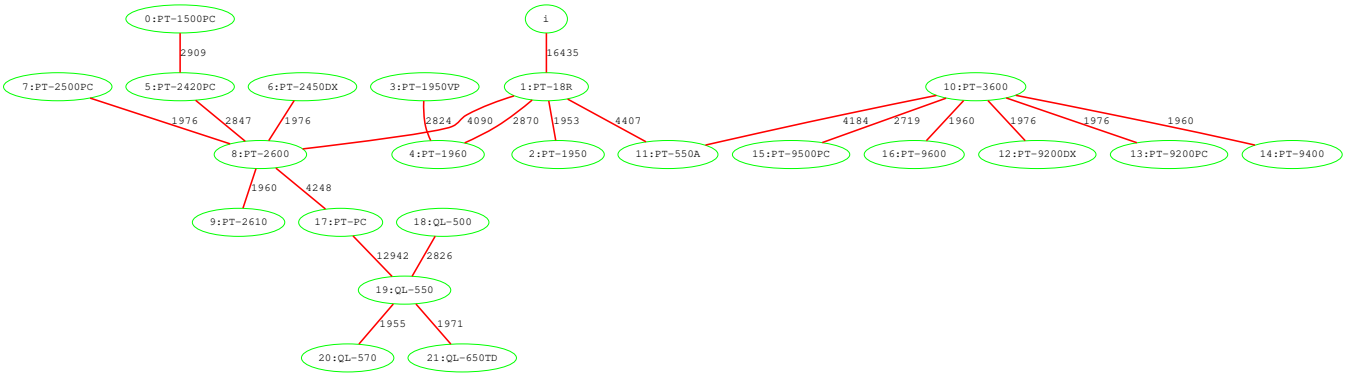


Figure 6: The minimum spanning tree of the diff-distances graph for the pto dataset (ptouch, 22 files). The details are similar to those for Fig. 5.

collection. It bears recalling that this comparison is not among equivalent alternatives, in that *dt* and *xdt* provide the organization advantages that the other schemes do not. Nevertheless, reducing the time needed by the difference-based schemes on large collections is an important area for further work.

One option for reducing the time required to compute a diff-distance graph is computing the diff-distances only for a fraction of document-pairs in a probabilistic manner. In more detail, for each pair of vertices (files), the diff is computed with a given probability. Edges corresponding to file-pairs whose diffs were not computed have their weight set to ∞. Fig. 12 summarizes the results of such a method on the pto collection. The results suggest that a substantial fraction fo diff computations may be skipped without adversely affecting the resulting compression ratio. Figs. 13 and 14 summarize analogous results for the foo and dym collection, and also suggest that a substantial fraction of diff computations may be omitted without adverse effects on the compression ratio.

5 RELATED WORK

There is a vast body of work, spanning several decades, on compression [15]. Much of it is applicable to parts of the general problem addressed here, and to the specific variant (based on differencing) that is the focus. For instance, well-established methods such as LZMA are applicable to the compression of inter-document differences (and indeed the *diboc* implementation uses it for this purpose). However, general purpose compression is not very effective on its own for organizing and compressing file collections for two main reasons: First, such methods pay little or no attention to the organization aspect of the problem, focusing instead on compression. Therefore, even if they may provide competitive compression, they do nothing to aid a human in understanding the relationships among files in a collection. Second, many such methods are not very effective at capitalizing on inter-document similarities, even from the narrower compression perspective. The latter observation has led to relatively recent work such as *FemtoZip* [20]. There is also a well developed body of work focusing on differencing, including the classic methods [12], later methods [18] as implemented in

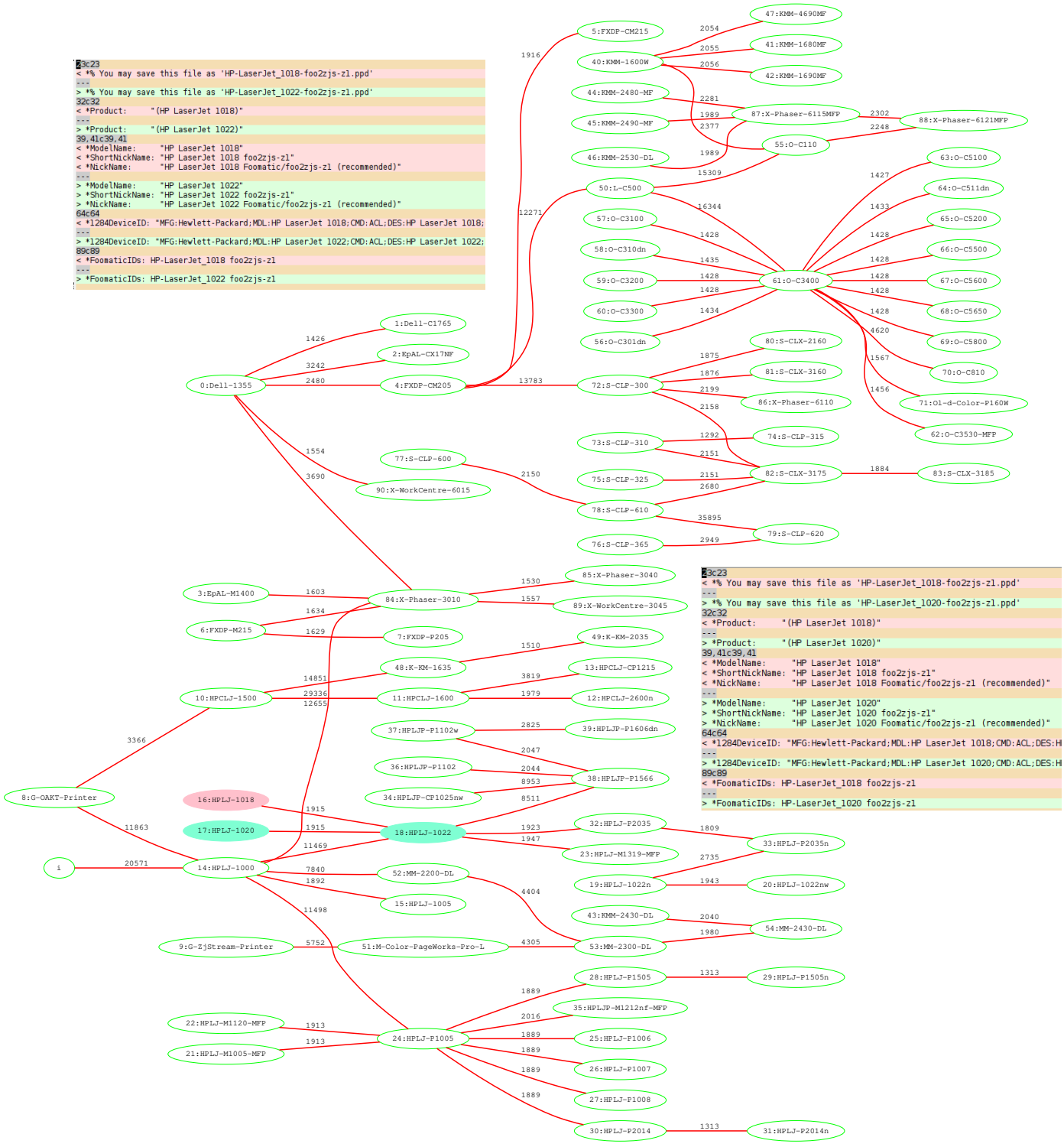


Figure 7: The minimum spanning tree of the diff-distances graph for the `foo` dataset (`foo2zjs`, 91 files). The details are similar to those for Fig. 5.

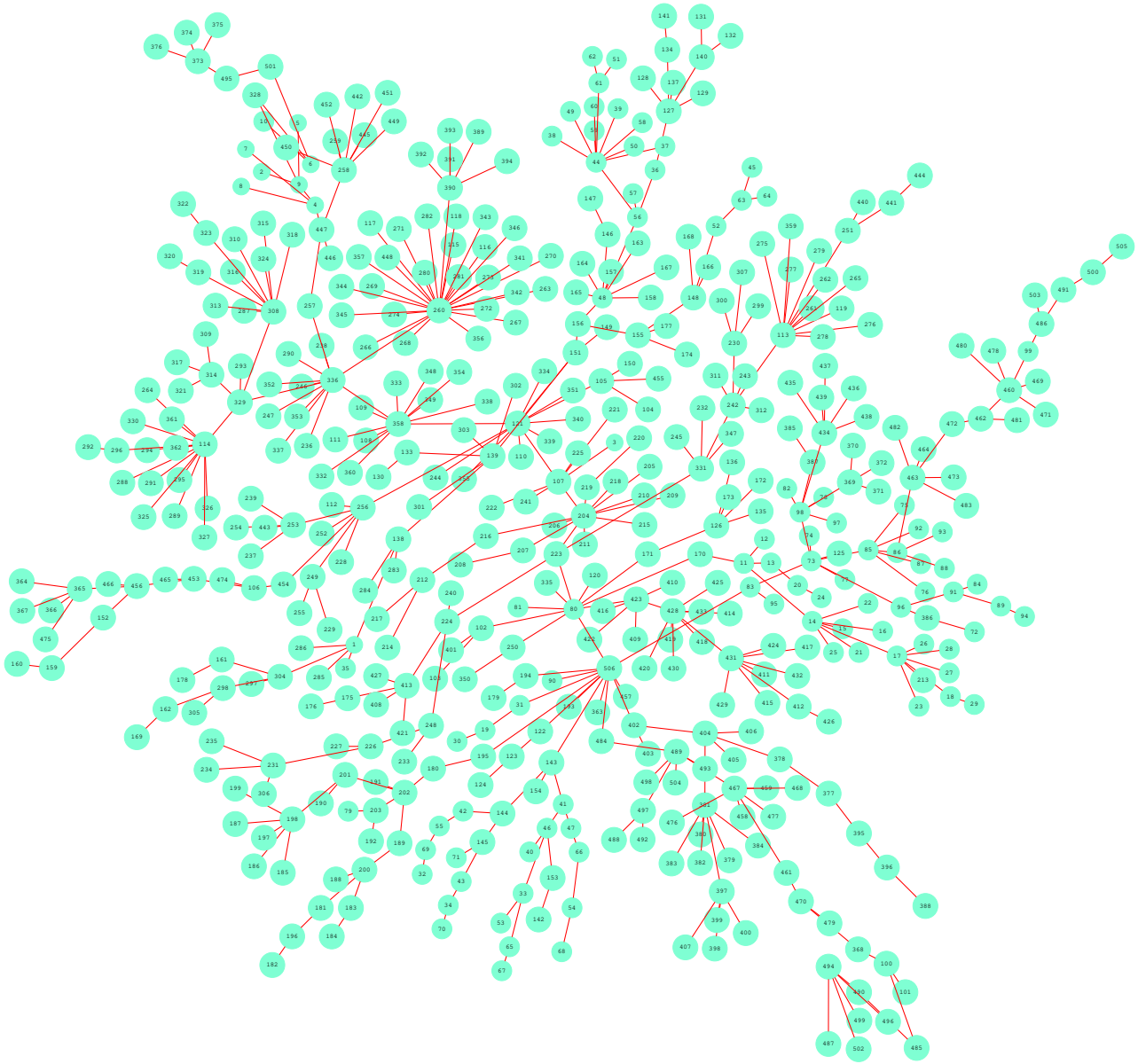


Figure 8: The minimum spanning tree of the diff-distances graph for the esc dataset (escpr, 505 files). The details are similar to those for Fig. 5 but all labels are elided (for presentation purposes here) except for numeric vertex identifiers.

popular utilities [16], and more recent work focusing on structured differencing [5] Recent work has combined the ideas on collection-based compression from FemtoZip with explicit differencing to achieve improved compression of document collections such as those studied in this paper [6].

Compression using shared dictionaries and/or differences, albeit at a lower level of abstraction that is tailored more for machines than humans, is an enduring idea that has several expressions in efforts such as Shared Dictionary Compression over HTTP (SDCH) [4] and VCDIFF [2, 14]. In the publish-subscribe environment, recent work has explored the use shared dictionaries for compressing message streams [9, 10].

The current *diboc* implementation uses *XZ* for compressed diffs. It would be worth considering alternatives such as VCDIFF [14] and FemtoZip [20], especially for collections that are larger, both by file counts and file sizes. The current implementation already favors faster decompression at the expense of slower compression. However, in environments in which file-extraction times are critical, it would be worth considering the *ZStandard* algorithm [7] as an alternative to the LZMA algorithm used by *XZ*, gaining faster

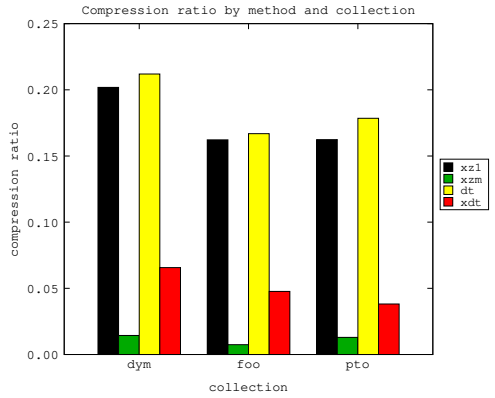


Figure 9: Compression ratio of four methods (colored bars, identified by legend) for three collections (bar clusters, identified by tags as in in Table 1).

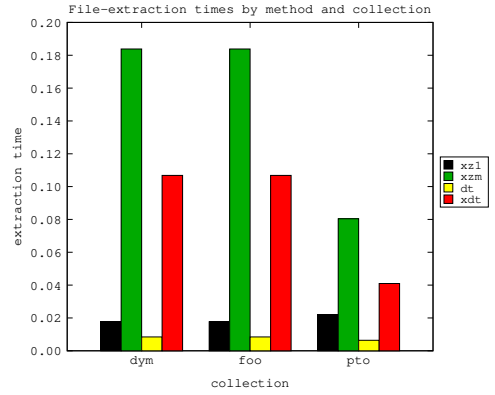


Figure 10: Extraction times (seconds) of four methods (colored bars, identified by legend) for three collections (bar clusters, identified by tags as in in Table 1).

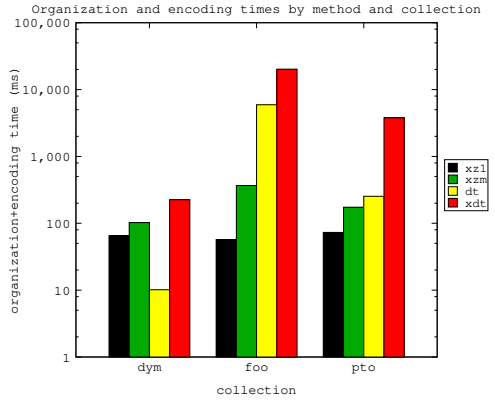


Figure 11: Collection organization and encoding times (milliseconds, on *logarithmic scale*) of four methods (colored bars, identified by legend) for three collections (bar clusters, identified by tags as in in Table 1).

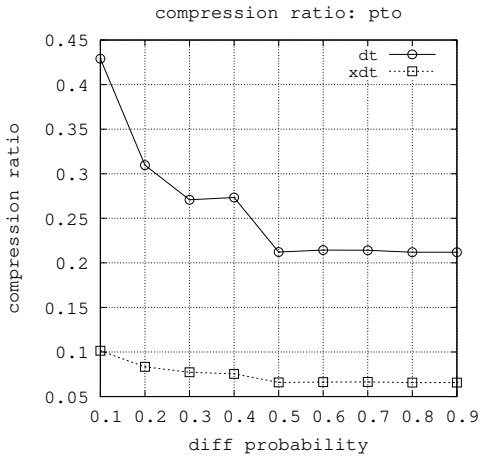


Figure 12: The effect of varying the probability of diff-computation (horizontal axis) on the compression ratio (vertical axis, ratio of compressed size to original size) for the uncompressed (`dt`) and compressed (`xdt`) diff methods on the `pto` dataset of Table 1.

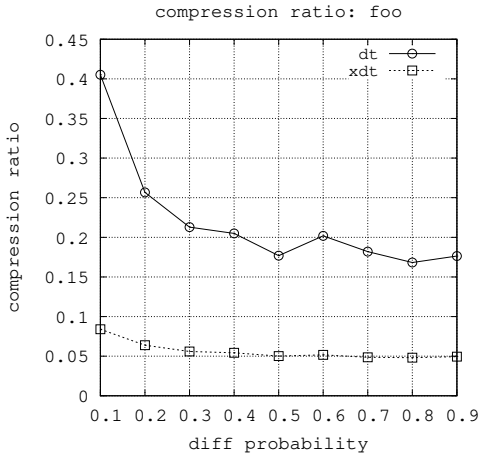


Figure 13: The effect of varying the probability of diff-computation (horizontal axis) on the compression ratio (vertical axis, ratio of compressed size to original size) for the uncompressed (`dt`) and compressed (`xdt`) diff methods on the `foo` dataset of Table 1.

decompression at the expense of potentially poorer compression ratios. Other related options include *Brotli* [1].

6 CONCLUSION

Collections of textual files, or documents, that are related in some manner, such as purpose, structure, author, or language, are common in diverse environments. The task of compressing such collections, with the usual goals of favorable compression ratios and low compression and decompression times, has received some attention in prior work, although surprisingly little in comparison with the larger body of work on compression in general. Similarly,

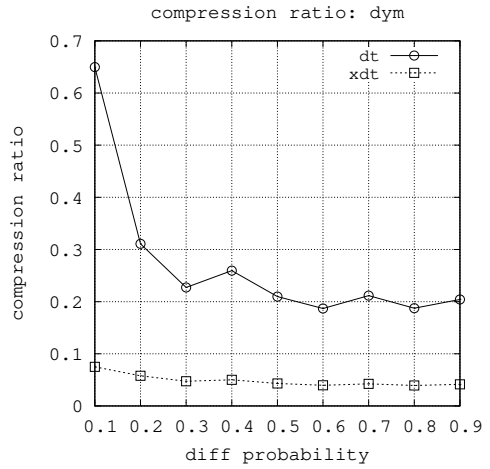


Figure 14: The effect of varying the probability of diff-computation (horizontal axis) on the compression ratio (vertical axis, ratio of compressed size to original size) for the uncompressed (`dt`) and compressed (`xdt`) diff methods on the `dym` dataset of Table 1.

the task of automatically organizing such collections by clustering, classification, etc., has also received significant attention. However, the task of simultaneously compressing and organizing these collections, with an equitable balance between the two desiderata, has not received much attention despite its practical significance [13].

This paper has motivated the need for such simultaneous and balanced compression and organization using a corpus of file-collections that is in widespread use and of practical significance. It has formulated the problem both in general and in a more specific variant based on inter-document differences. Using a well known and commonly used family of differencing tools provides the significant advantage of easing human comprehension of file collections because researchers and practitioners alike are likely to be familiar with interpreting differences expressed in this manner. After some preliminary options are settled, this specific variant of the problem conveniently reduces to the well known problem of computing a minimum (weight) spanning tree of an (edge) weighted graph.

The paper has outlined the implementation of the *diboc* prototype system based on these ideas. An experimental evaluation uses the corpus of PPD files used by the CUPS printing system, which is in widespread use. Using the output of *diboc* on several collections in this corpus, the paper has provided anecdotal evidence of the value of the methods and implementation for human-understandable organization of file collections. The ability of the system to discover and propose plausible structure in file collections, and to guide and enable a human to selectively examine the files and differences of importance, is notable. As well, the implementation, despite being very unoptimized, provides competitive performance on the compression-related metrics such as compression ratio. A weakness, and an area of ongoing work, is the time required for computing the diff-distances graph. The current method and implementation has $\Theta(n^2)$ time and space complexity, which prohibits scaling to very large collections. Nevertheless, the *diboc* implementation is already useful when applied to several real file collections of practical significance, as illustrated by the experimental study.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation. It benefited from the reviewers' comments.

REFERENCES

[1] Jyrki Alakuijala and Zoltan Szabadka. 2016. Brotli Compressed Data Format. Internet Engineering Task Force (IETF). Request for Comments 7932. https://tools.ietf.org/html/rfc7932

[2] Jon Bentley and Douglas McIlroy. 1999. Data Compression Using Long Common Strings. In *Proceedings of the IEEE Data Compression Conference (DCC'99)*. IEEE Computer Society, 287–295. https://doi.org/10.1109/DCC.1999.755678

[3] Per Bothner. 1998. Kawa: Compiling Dynamic Languages to the Java VM. In *Proceedings of the USENIX Annual Technical Conference, FREENIX Track* (New Orleans, Louisiana). USENIX Association, Berkeley, CA, USA, 41–41. http://dl.acm.org/citation.cfm?id=1268256.1268297

[4] Jon Butler, Wei-Hsin Lee, Bryan McQuade, and Kenneth Mixter. 2008. *A Proposal for Shared Dictionary Compression over HTTP.* Technical Report. Google, Inc. https://lists.w3.org/Archives/Public/ietf-http-wg/2008JulSep/att-0441/Shared_Dictionary_Compression_over_HTTP.pdf IETF-HTTP-WG mailing-list message.

[5] Sudarshan S. Chawathe. 2005. Differencing Data Streams. In *Proceedings of the 9th International Database Engineering and Applications Symposium (IDEAS '05)*. IEEE Computer Society, USA, 273–284. https://doi.org/10.1109/IDEAS.2005.21

[6] Sudarshan S. Chawathe. 2020. Efficient File Collections for Embedded Devices. In *Proceedings of the 8th Workshop on Communications in Critical Embedded Systems (WoCCES 2020)*. IEEE, Rennes, France.

[7] Yann Collet and Murray S. Kucherawy. 2018. Zstandard Compression and the application/zstd Media Type. Internet Engineering Task Force (IETF). Request for Comments 8478. https://tools.ietf.org/html/rfc8478

[8] Lasse Collin et al. 2020. The Tukaani Project: XZ Utils. https://tukaani.org/xz/.

[9] Christoph Doblander, Tanuj Ghinaiya, Kaiwen Zhang, and Hans-Arno Jacobsen. 2016. Shared Dictionary Compression in Publish/Subscribe Systems. In *Proceedings of the 10th ACM International Conference on Distributed and Event-Based Systems* (Irvine, California) *(DEBS '16)*. Association for Computing Machinery, New York, NY, USA, 117–124. https://doi.org/10.1145/2933267.2933308

[10] Christoph Doblander, Arash Khatayee, and Hans-Arno Jacobsen. 2018. PreDict: Predictive Dictionary Maintenance for Message Compression in Publish/Subscribe. In *Proceedings of the 19th International Middleware Conference* (Rennes, France) *(Middleware '18)*. Association for Computing Machinery, New York, NY, USA, 174–186. https://doi.org/10.1145/3274808.3274822

[11] Emden R. Gansner and Stephen C. North. 2000. An open graph visualization system and its applications to software engineering. *Software: Practice and Experience* 30, 11 (2000), 1203–1233. https://doi.org/10.1002/1097-024X(200009)30:11<1203::AID-SPE338>3.0.CO;2-N

[12] James W. Hunt and Thomas G. Szymanski. 1977. A Fast Algorithm for Computing Longest Common Subsequences. *Commun. ACM* 20, 5 (May 1977), 350–353. https://doi.org/10.1145/359581.359603

[13] Till Kamppeter et al. 2020. pyppd: A CUPS PostScript Printer Driver's compressor and generator. https://github.com/OpenPrinting/pyppd.

[14] David G. Korn, Joshua P. MacDonald, Jeffrey C. Mogul, and Kiem-Phong Vo. 2002. The VCDIFF Generic Differencing and Compression Data Format. Internet Engineering Task Force (IETF). Request for Comments 3284.. https://tools.ietf.org/html/rfc3284

[15] Debra A. Lelewer and Daniel S. Hirschberg. 1987. Data Compression. *Comput. Surveys* 19, 3 (Sept. 1987), 261âĂŞ296. https://doi.org/10.1145/45072.45074

[16] David MacKenzie, Paul Eggert, and Richard Stallman. 2018. GNU diffutils—Comparing and Merging Files. Free Software Foundation. https://www.gnu.org/software/diffutils/manual/diffutils.html Manual for diffutils version 3.7.

[17] Dimitrios Michail, Joris Kinable, Barak Naveh, and John V. Sichi. 2020. JGraphT—A Java Library for Graph Data Structures and Algorithms. *ACM Trans. Math. Softw.* 46, 2, Article 16 (May 2020), 29 pages. https://doi.org/10.1145/3381449

[18] Eugene W. Myers. 1986. AnO(ND) difference algorithm and its variations. *Algorithmica* 1, 1 (1986), 251–266. https://doi.org/10.1007/BF01840446

[19] java-diff-utils contributors on GitHub. 2020. Java Diff Utils. https://github.com/java-diff-utils/java-diff-utils/.

[20] G. Toubassi. 2012. How FemtoZip Works (In Painful Detail). Software system documentation. https://github.com/gtoubassi/femtozip/wiki/How-FemtoZip-Works-(In-Painful-Detail).

Local Connectivity in Centroid Clustering

Deepak P
Queen's University Belfast, UK
deepaksp@acm.org

ABSTRACT

Clustering is a fundamental task in unsupervised learning, one that targets to group a dataset into clusters of similar objects. There has been recent interest in embedding normative considerations around fairness within clustering formulations. In this paper, we propose 'local connectivity' as a crucial factor in assessing membership desert in centroid clustering. We use local connectivity to refer to the support offered by the local neighborhood of an object towards supporting its membership to the cluster in question. We motivate the need to consider local connectivity of objects in cluster assignment, and provide ways to quantify local connectivity in a given clustering. We then exploit concepts from density-based clustering and devise LOFKM, a clustering method that seeks to deepen local connectivity in clustering outputs, while staying within the framework of centroid clustering. Through an empirical evaluation over real-world datasets, we illustrate that LOFKM achieves notable improvements in local connectivity at reasonable costs to clustering quality, illustrating the effectiveness of the method.

CCS CONCEPTS

• **Information systems** → **Clustering**.

KEYWORDS

Clustering, Local Connectivity, Normative Considerations

1 INTRODUCTION

Clustering [12] has been a popular task in unsupervised learning. Clustering involves grouping a dataset of objects into a number of groups such that objects that are highly similar to one another are more likely to find themselves assigned to the same group, and vice versa. Clustering algorithms fall into one of many families, of which partitional and hierarchical algorithms are two main streams. Partitional clustering, arguably the more popular stream, considers grouping the dataset into a number of disjoint sets. The pioneering work in this family, K-Means clustering, dates back to the 1960s [16]. K-Means clustering is a partitional clustering algorithm that additionally outputs a prototypical object to *'represent'* each cluster, which happens to simply be the cluster *centroid* within the basic K-Means formulation. The centroid output is often seen as very useful for scenarios such as for manual perusal to ascertain cluster characteristics, resulting in this paradigm of *'centroid clustering'* [23] attracting much research interest. In alternative formulations within the centroid clustering paradigm, the prototypical object is set to be the medoid, which is a dataset object that is most centrally positioned; this is referred to as K-medoids [20] clustering or PAM[1]. 50+ years since K-Means, the basic K-Means formulation is still used widely and continues to inspire much clustering research [11]. The second popular family of clustering algorithms, that of hierarchical clustering, focuses on generating a hierarchy of clusters from which clusterings of differing granularities can be extracted. An early survey of hierarchical clustering methods appears at [17]. Our focus in this paper is within the task of centroid clustering.

1.1 Membership Desert in Centroid Clustering

In this paper, we problematize the notion of *cluster membership* in centroid clustering from a conceptual and normative perspective. Our work is situated within the context of recent interest in fairness and ethics in machine learning (e.g., [15]), which focuses on embedding normative principles within data science algorithms in order to align them better with values in the modern society. In particular, we consider the question of *membership desert*, or what it means for an object to be deserving of being a member of a cluster, or a cluster to be deserving of containing a data object. Desert in philosophical literature[2] refers to the condition of being deserving of something; a detailed exposition of philosophical debate on the topic can be found within a topical encyclopaedia from Stanford[3]. K-Means and most other formulations that build upon it have used a fairly simple notion of membership desert; that an object be assigned to the cluster to whose prototype it is most proximal, according to a task-relevant notion of similarity. While this simple notion makes intuitive sense as well as enables convenient optimization, it admits unintuitive outcomes as we will see later.

There have been two recent works in re-considering membership desert in centroid clustering, both within the umbrella of research in fair machine learning. The first work [8] considers a notion of *collective desert* to blend in with the K-Means framework, whereby a reasonably large set of objects is considered to be deserving of their own cluster as long as they are collectively proximal to one another. The second work [19] considers the distance-to-centroid as a cost of abstraction incurred by objects in the dataset, and strives to achieve a fair distribution of the cost of abstraction across objects. We will discuss these in detail in a later section. In this work, we consider advancing a third distinct normative consideration in membership desert, that of *local connectivity*. At the high level, we consider the membership desert associated with an object-cluster pair as being

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410601

[1] https://en.wikipedia.org/wiki/K-medoids
[2] https://en.wikipedia.org/wiki/Desert_(philosophy)
[3] https://plato.stanford.edu/entries/desert/

intimately related to the extent of the object's neighbors' affinity towards the cluster in question.

1.2 Our Contributions

In what may be seen as a contrast to conventional research narratives within data analytics, our work is centered on advancing a particular normative consideration as opposed to a technological challenge. This is in line with recent work on fairness and ethics in AI, which have mostly appeared within data analytics avenues as well (e.g., [1, 5, 8]). Our contribution by way of this work is three-fold:

- **Local Connectivity as Membership Desert:** We develop an argument for considering *local connectivity* as a notion of membership desert in centroid clustering. Building upon this argument, we develop quantitative metrics to evaluate the extent to which local connectivity is being adhered to, within a clustering.
- **LOFKM:** We develop a simple centroid clustering formulation, *LOFKM*, drawing inspiration from both centroid clustering and density-based clustering, that deepens local connectivity in clustering outputs.
- **Evaluation:** Through an empirical evaluation over multiple real-world datasets, we illustrate that LOFKM is able to significantly improve alignment with local connectivity considerations at reasonable costs to clustering quality.

Roadmap: We start by considering related work in Section 2, followed by an overview of membership desert in Section 3. This is followed by Section 4 where we describe local connectivity as a distinct notion of membership desert and ways of quantifying it for a given clustering. Section 5 outlines a simple method for enhancing local connectivity in centroid clustering, codenamed *LOFKM*. This is followed by our experimental evaluation in Section 6, a brief discussion in Section 7 and conclusions in Section 8.

2 RELATED WORK

Given that our work advances a local neighborhood based normative consideration in clustering, we briefly summarize related work from (i) fair clustering, and (ii) local neighborhood estimations from the density-based clustering family.

2.1 Fair Clustering

There has been an emerging interest in fair clustering. Among the two notions of fairness, *individual* and *group* fairness [6], fair clustering has largely seen explorations on the latter. Group fairness involves ensuring cluster-level representational parity of sensitive groups defined on attributes such as gender, race, ethnicity and marital status. This literature, initiated by a work on ingenious dataset pre-processing [9], has seen work on embedding fairness within the optimization [1] as well as in post-processing [5]. These also differ in the number of types of sensitive attributes that they admit. An overview of recent work on group-fair clustering appears in [1] (Ref. Table 1 therein). Research into individual fairness in clustering has a flavour of considering membership desert as the focus question; being pertinent to our work, we discuss this in detail in Section 3.

2.2 Local Neighborhood and Clustering

Local neighborhood of objects has been the core consideration in work on density-based clustering, a field pioneered by the DBSCAN clustering algorithm [10], followed by OPTICS [2]. In our work, we will make use of a work that extends concepts from density-based clustering in order to identify the outlierness of dataset objects, called *Local Outlier Factor* (LOF) [7]. The structure of LOF relies on quantifying the *local density* around an object. The local density around an object is inversely related to the average *reachability* of the object to its k nearest neighbors; with *reachability* being a lower-bounded version of distance between the objects. The local density around an object's neighbors is then contrasted with the object's own local density to arrive at the LOF, which is a non-negative real number. $LOF > 1$ ($LOF < 1$) is achieved by objects whose neighbors are in neighborhoods that are denser (sparser) than it's own, with $LOF = 1$ indicating a good match between respective densities. Objects with high values of LOF, especially $LOF >> 1$, are considered density-based outliers, due to their (relative) lack of closeby neighbors. Over the past two decades, LOF has evolved to being a very popular outlier detection method, continuously inspiring systems work on improving efficiency (e.g.,a recent *fast LOF* work appears in [3]), arguably adorning a place in the outlier detection literature only next to the analogous status of K-Means within clustering literature.

3 BACKGROUND: MEMBERSHIP DESERT IN CENTROID CLUSTERING

Following up from Section 1.1, we now cover more background on the notion of membership desert in K-Means, and recent fairness oriented re-considerations of the notion.

3.1 Critiquing K-Means' Membership Desert

Let us start with looking at the simple notion of membership desert used in K-Means, that an object deserves to be assigned to the cluster whose prototype[4] it is most proximal to, proximity measured under a domain-specific notion of (dis)similarity that is deemed relevant to the clustering task. *First*, consider the case of two clusters, A and B. Now, let an object X_1 be at a distance of 3 and 5 units from the prototypes of A and B respectively, as shown roughly in the first illustration in Fig 1. For another object X_2, also shown in the illustration, let the distances be 8 and 6 respectively. The simple K-Means (*argmin*) heuristic does the following assignment: $X_1 \in A$ and $X_2 \in B$. It may be noted that while considering proximity as membership desert as in K-Means, X_1 may be considered more deserving of being assigned to B than X_2 is to B; this is so since $dist(X_1, B) < dist(X_2, B)$. However, the K-Means assignment is in conflict with this observation, due to the higher degree of proximity of X_1 to A. *Second*, consider a scenario with respect to the trio, X_1 in relation to A and B, as shown in the right-side in Figure 1. Let B be a naturally bigger and denser cluster with significant number of data objects within 6 units of distance of it. On the other hand, let A be a small cluster with most of its members being within 2 units of distance around its prototype. In this setting, despite $dist(X_1, A) < dist(X_1, B)$, X_1 may be thought of as deserving of

[4] we use prototype and centroid interchangeably

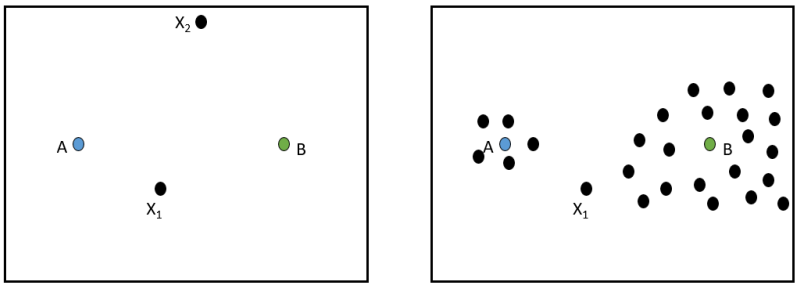


Figure 1: Two Cases for Section 3.1: Rough Illustration

being located within B since it is in the company of the large mass of points stretching to the proximity of B. This intuitive notion of membership desert also conflicts with the cluster assignment that K-Means does. In fact, this is also an fallout of a fundamental design assumption in K-Means, that clusters be modelled as being modeled as Voronoi cells. *While we do *not* argue that the K-Means choice is inferior to an alternative available choice, it may be seen that there are intuitive opportunities to critique the simple membership desert mechanism in K-Means, and that the choice of most proximal centroid is not the only natural choice.* It is also noteworthy that membership assignment is not a final K-Means step, making it not entirely appropriate to consider it in isolation as we have done so far. The cluster assignment step is interleaved with the centroid learning step, leading to an interplay of effects of each other.

3.2 Fairness-orientated Notions of Membership

As outlined earlier, there are two recent papers, that motivate different considerations in cluster membership assignment.

3.2.1 Proportionality [8] or Collective Desert in Cluster Membership. K-Means uses a parameter, the number of expected clusters in the output, commonly denoted as K. Thus, on an average, there are (n/K) objects in a K-Means cluster. Proportionality, a concept the authors propose, is the notion that if one can find a set of $\lceil n/K \rceil$ data objects that collectively prefer the same candidate centroid in lieu of their current assignments (which involve different centroids/clusters), they deserve a cluster of their own centered at the candidate centroid that they collectively prefer. A clustering would be regarded as violating proportionality if it involves denying this set of $\lceil n/K \rceil$ objects their own cluster that they deserve. They develop algorithms that generate *proportionally fair* clusterings, those that do not violate proportionality.

3.2.2 Representativity Fairness [19]. A recent work considers human-in-the-loop analytics pipelines where each cluster centroid is perused in order to arrive at a single decision for all objects in the cluster. Within such pipelines and even more generally, objects that are far away from their assigned cluster centroids suffer a higher *'representativity cost'* from the cluster-level abstraction of the dataset provided by the clustering. RFKM, the proposed method, seeks to level off this object-level cost across the objects in the dataset, and move towards what is called *representativity fairness*. Operationally, it considers re-engineering the K-Means steps in a way that chances of proximity violations such as those in the first example in Section 3.1 are reduced.

4 LOCAL CONNECTIVITY AND MEMBERSHIP DESERT

4.1 Motivation

We first consider *local connectivity* as a concept and its relevance to membership desert in centroid clustering. Consider three motivating scenarios in Fig. 2. In each of these figures, the middle point is the designated cluster prototype for the blue cluster; in other words, we have zoomed in on the blue cluster prototype and excluded other points in the dataset (including those from blue or other clusters) from view. The other blue colored points are assigned to be part of the blue cluster, and the red colored points in Fig. 2(a) are part of a different (red) cluster. In each of these figures, we would like to consider the status of the black colored object, and how well it deserves to be part of the blue cluster, and thus to being *'represented'* by the blue cluster's prototype in the cluster-level abstraction.

Fig 2(a) has the corresponding black object being closest to the blue cluster prototype among all three scenarios. However, its local neighborhood (think of it as perhaps the closest few data objects to itself) is largely from the red cluster. Intuitively, this makes it reasonable to argue that despite the proximity, the black object in Fig 2(a) is limited in how well it deserves to be part of the blue cluster; in other words, its membership desert to the blue cluster comes under question. Now, consider the scenario in Fig 2(b). The black object, while not as proximal as in the case of Fig. 2(a), is quite well connected to the blue cluster given that it has an *'pull'* from its local neighborhood towards the blue cluster. This makes it more deserving of membership to the blue cluster. Lastly, consider Fig 2(c) where the black object is tucked into a corner within a sparse region of the space. It has a reasonable claim to membership in the blue cluster, due to its nearest neighbors being blue (despite them being quite far from itself); however, the strength of the claim is dented by its distance to the blue cluster prototype. In summary, we observe the following:

- Fig 2(a): Despite proximity, the membership desert of the black object to the blue cluster is limited due to the local neighborhood being red.
- Fig 2(b): The black object is most deserving to be part of the blue cluster due to high local connectivity within the blue cluster and reasonable proximity to the blue cluster prototype.

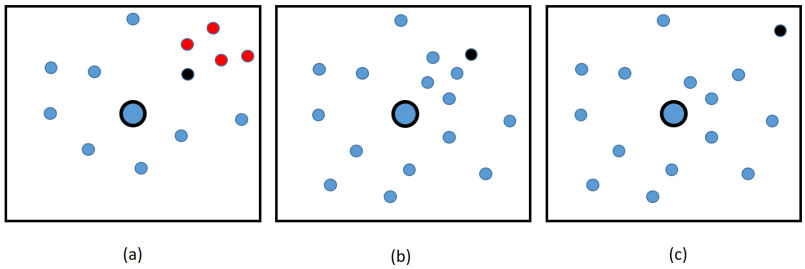


Figure 2: Local Connectivity: Motivating Scenarios (best viewed in color)

- Fig 2(c): The black object may be considered as reasonably deserving of blue cluster membership, even though its distance from the blue cluster prototype reduces the strength of the claim.

In other words, these illustrative scenarios offer different trade-offs between the pull towards the blue cluster prototype offered by *local connectivity* and *proximity*. These, we hope, illustrates that local neighborhood connectivity to the cluster in question is a fairly crucial factor in assessing membership desert. Though we have used abstract examples to motivate local connectivity, this has real-world implications wherever clustering is used for consequential tasks; for a simple example, consider centroid clustering being used for *facility location* to determine locations of service facilities (e.g., post offices or hospitals) with people represented using their geographic co-ordinates. In facility location, assigning a person to a facility (located at a centroid) towards which she has few local neighbors may be seen as unjust as well as a decision that undermines social solidarity.

While K-Means is evidently not directly accommodative of local connectivity considerations due to using proximity in cluster assignment, the family of density based clustering algorithms pioneered by DBSCAN [10, 22] makes local neighborhood a prime consideration in forming clusters. However, the density-based clustering family does not offer a convenient prototype for each cluster, and is thus limited in its applicability to human-in-the-loop pipelines such as those outlined in [19]. In particular, density-based clusterings could yield non-convex clusters, where the centroid computed over cluster objects could be situated outside the natural boundaries of the cluster. Our method, as we will see, will leverage concepts from local neighborhood assessments from the density-based clustering family, and use that within the framework of centroid clustering inspired by K-Means.

4.2 Quantifying Local Connectivity

Local connectivity in cluster membership desert, as illustrated in the previous section, can be thought of as: *how well the local neighborhood of the data object supports its membership to the cluster in question*. We now consider quantifying local connectivity at the object level, which will be aggregated to the level of different clusters in order to arrive at a measure of how well local connectivity is adhered to, in a given clustering. This quantification would form an evaluation metric for assessing local connectivity in clustering.

Consider an object whose cluster-specific local neighborhood is conceptualized as the set of its t nearest neighbors (we use t instead of the conventional k to avoid conflict with the K in K-Means) within the cluster in question. We would like the t nearest neighbors to comprise objects that:

- **Offer a Cluster Pull:** We would like the neighbors to offer a pull in the direction towards the cluster prototype. While *pull* is admittedly an informal word, we believe it is fairly straightforward to interpret the meaning. To illustrate this notion, observe that the local neighborhood in Fig 2(a) was largely red objects which may be seen as pulling the object towards the red cluster. This is in sharp contrast with the local neighborhood pull towards the blue cluster in Fig 2(b).
- **Are Proximal to the Object:** Even if the t nearest neighbors are towards the cluster prototype and can be seen as offering a pull, such a pull is meaningless unless the neighbors are proximal to the object in question. For example, consider Fig 2(c) where the neighbors of the black object are all towards the blue cluster. However, the appeal of this pull is dented by the fact that the neighbors are quite distant from the black object.

We now quantify the above desired characteristics in the form of a quantitative measure, for a given clustering. Let X be the data object in question, and C be the cluster prototype to whom the local connectivity strength is to be estimated. The dataset of objects involved in the clustering is denoted as $\mathcal{X}$. Given our interest in quantifying the pull towards the cluster prototype, we first identify the set of t nearest neighbors of X that are both: (i) members of the cluster in question i.e., C, and (ii) lie *in between* X and the cluster prototype for C. This set is denoted as $N_t^C(X)$:

$$N_t^C(X) = \underset{S \subseteq C \wedge Satisfies(S,X,C) \wedge |S|=t}{\arg\min} \sum_{s \in S} dist(s, X) \quad (1)$$

where:

$$Satisfies(S, X, C) = \bigwedge_{s \in S} (dist(s, C) \leq dist(X, C)) \wedge (dist(X, s) < dist(X, C)) \quad (2)$$

$Satisfies(.,.,.)$ enforces the condition that objects in $N_t^C(X)$ fall in between C and X through a distance check; the first distance condition checks whether each element s is closer to C, and the second checks whether it is on the *'same side'* of C as X is. Among objects that satisfy these conditions, t of them that are most proximal to X are chosen to form the set $N_t^C(X)$. It may be noted that in cases where there are not enough objects that satisfy the eligibility

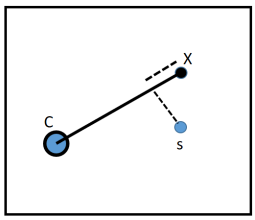


Figure 3: Quantifying Local Conenctivity Illustration

condition, $|N_t^C(X)|$ may be less than t. This is likely to happen when C is very close to X; we will outline its implications later.

Our interest is now in assessing how well objects in $N_t^C(X)$ adhere to the *pull* and *proximity* heuristics outlined above. We use a simple geometric intuition in order to quantify these. Consider Figure 3 where X is the black object and C is the big blue encircled object, as before. The small blue object is $s \in N_t^C(X)$. Consider the line joining X and C and s shown as being projected on to the line. The *pull* heuristic would prefer the dotted line indicating the projection of s to the line to be *as short as possible* since that would direct the pull offered by s to be aligned towards C. The *proximity* heuristic, on the other hand, would prefer s to be as close as possible to X, thus preferring that both the dotted lines be as short as possible. We would additionally like the local connectivity to be comparable across different data objects in $\mathcal{X}$. Thus, we measure the two distances indirectly in relation to the distance between X and C, as two measures, *Deviation* (Dev) and *Normalized Distance* (ND), as follows:

$$Dev(X, C, s) = \frac{dist(C, s) + dist(s, X)}{dist(X, C)} - 1.0 \tag{3}$$

$$ND(X, C, s) = \frac{dist(X, s)}{dist(C, s) + dist(s, X)} \tag{4}$$

$Dev(X, C, s)$ would evaluate to 0.0 when s falls directly on the line connecting X and C, since that would ensure that $dist(C, s) + dist(s, X) = dist(X, C)$. $Dev(X, C, s)$ increases the more s deviates from that line, leading to its name. $ND(X, C, s)$ on the other hand, measures the distance between X and s as a fraction of the distance between X and C through s. Thus, $ND(.,.,.)$, unlike $Dev(.,.,.)$ is directly related to the length of both dotted lines in Fig 3. Since we would like both of these measures to be numerically small (≈ 0), we would like to minimize the product of these, which we call as the *local connectivity disagreement* measure:

$$LCD(X, C, s) = Dev(X, C, s) \times ND(X, C, s) \tag{5}$$

Higher values of $LCD()$ denote lower levels of *local connectivity* offered by s to support the membership desert for the pair X, C. This disagreement may be aggregated across all objects in $N_t^C(X)$ to arrive at an object level estimate:

$$LCD(X, C) = \sum_{s \in N_t^C(X)} LCD(X, C, s) \tag{6}$$

When $|N_t^C(X)| < t$, the LCD would be correspondingly lower since there are fewer objects to sum over. Since we expect $|N_t^C(X)| < t$ to happen when X is already very close to C, this translates to an alternative route to reduce LCD for such objects; in addition to improving local connectivity by way of neighbors' positions, *LCD can also be improved (i.e., numerically reduced) through enhanced proximity between objects and their cluster prototypes, which would lead to smaller* $|N_t^C(X)|$. Among objects in the cluster C, some may have high LCDs and some may have lower values for LCD. Towards assessing a cluster, consider using the average of the LCDs across all objects as an aggregate measure. This would enable a small set of objects with very shallow local connectivity (i.e., high LCD scores since LCD measures disagreement) to be ignored due to being compensated by a large number of low LCD scores across other objects in the cluster. This may be considered undesirable in the face of the high importance accorded to the concern for the most disadvantaged, such as in the very popular stream of Rawlsian notions of fairness [13]. Motivated by such considerations, we accord the cluster with an LCD value computed as the highest LCD (i.e., lowest local connectivity, since LCD measures disagreement) among its objects:

$$LCD(C) = \max_{X \in C} LCD(X, C) \tag{7}$$

A clustering of a dataset would produce multiple clusters, since a clustering defines a partitioning. In order to arrive at a dataset-level measure of connectivity offered by a clustering, we would need an aggregate statistic at the dataset level. As in the case above, we would like to ensure that no cluster suffers from bad local connectivity, making the highest LCD among clusters a natural measure to minimize. We call this *MaxLCD*. Additionally, We would also like to minimize LCD across all clusters, making *AvgLCD* a very pertinent measure.

$$MaxLCD(\mathcal{C}) = \max_{C \in \mathcal{C}} LCD(C) \tag{8}$$

$$AvgLCD(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} LCD(C) \tag{9}$$

These are analogous to the construction of Max Wasserstein and Avg Wasserstein used in evaluation of fair clustering [1]. Thus, *MaxLCD* and *AvgLCD* offer quantifications of disagreement with local connectivity across the dataset, as manifested in the clustering $\mathcal{C}$. A good clustering would be one which, in addition to performing well on traditional clustering evaluation metrics such as *purity* and *silhoutte*, achieves *low values of MaxLCD and AvgLCD* (thus, high local connectivity).

4.3 Drawbacks of LCD Measures

While LCD measures are, we believe, a starting point for quantifying local connectivity, these are not free of shortcomings. We outline a few drawbacks, which could potentially point to ways of refining them to yield better metrics of local connectivity.

First, both $Dev(.)$ and $ND(.)$, which form the building blocks of LCD measures, rely on distances expressed as fractions of other distances. This makes them unable to be sensitive to variations in absolute distances. Consider the case of $Dev(.)$; when X and C are close to each other, even slight deviations of s from the straight line connecting them are amplified, with $dist(X, C)$ forming the denominator. Similarly, take the case of $ND(.)$; high values of $dist(C, s)$ push it towards 0.0 by providing a very high denominator. When X

and s are very far from C, even high values of $dist(X, s)$ could cause $ND(.) \approx 0$. Such cases make *LCD* less meaningful to quantify the connectivity of fringe objects that are far from cluster prototypes. Any attempts at addressing such absolute distance issues should also care to retain the comparability of the resultant metrics across objects in the dataset. *Second*, we have excluded neighbors of X that do *not* belong to the same cluster as C, from consideration in $N_t^C(X)$. This means that an object's neighbors' pull towards the assigned cluster is evaluated without regard to whether it has similar or stronger pulls towards other clusters. This, we believe, is a minor issue, since such stronger pulls towards a different cluster also would likely reduce cluster coherence in general. This means that any clustering that attempts to improve coherence of clusters in addition to local connectivity (such as our method, *LOFKM*, introduced later) would address this implicitly to some extent using the cluster coherence criterion.

The above two sets of drawbacks are not meant to be comprehensive but to serve to provide a flavour of the possibilities of improving upon LCD measures, and the challenges in those directions.

5 LOFKM: ENHANCING LOCAL CONNECTIVITY IN CLUSTERING

We have argued and motivated that local connectivity is a crucial factor in considering membership desert for an object to a cluster. Local neighborhood statistics has been extensively used in the stream of work on density-based clustering, initiated through the popular DBSCAN clustering method [10]. Density-based clustering has the ability to identify clusters that have non-convex shapes (e.g., can disambiguate star and crescent[5] as separate clusters) and overlapping convex shapes (e.g., can identify rings arranged concentrically as separate clusters). However, this ability comes at a cost; density-based clustering inherently lacks the possibility of choosing a meaningful representative prototype for a cluster (e.g., in the above cases, observe that the centroid would lie outside the cluster itself and would be meaningless as a prototype). Our method, LOFKM, is the result of an attempt to bring a density-based flavour within K-Means framework, in order to improve local conncectivity considerations.

Our design considerations are as follows:

- *Conceptual Simplicity:* We would like to retain the conceptual simplicity inherent in K-Means which has likely been at the core of it's widespread popularity. Additionally, we would like to bring in density-based concepts within it in a lucid manner.
- *Computational Convenience:* The task of clustering is a dataset-level optimization problem which has inherent complexities. This makes directly using local connectivity measures (e.g., LCD) within the optimization infeasible. Due to solving a computational task, computational convenience is also a significant consideration.

5.1 Towards a Method

As we have seen, *local connectivity* involves a relation between an object and a cluster prototype in the backdrop of the local neighborhood of the object *in the 'direction' of the cluster prototype*. It is important to note that the local neighborhood of an object is a property of its location within the similarity space provided by the pre-specified $[dataset, distance\ function]$ pair, and is in no way *'alterable'* to nudge clustering towards deepening local connectivity (or any other consideration, for that matter).

High *LOF* (Ref. Sec 2.2) objects are more likely to suffer from shallow local connectivity since their neighborhood is sparse; so the neighbors are unlikely to support their membership to any cluster by much. One way to enhance local connectivity would be through better *inlineness*, which would be to set cluster prototypes in such directions from high LOF objects within which they have many neighbors. This, however, would require a significantly different prototype construction, putting the conceptual simplicity of K-Means prototype estimation at risk. Yet another way would be to bring the cluster prototype towards such high *LOF* objects, which would enhance their connectivity through both support from neighborhood as well as lower $|N_t^C(X)|$. This route is amenable to exploration while staying within the framework of the K-Means clustering formulation, and forms the basis of our *LOFKM* method. However, it risks bringing down the compactness of the cluster, which is a factor that would have repercussions on other metrics such as cluster purity and silhoutte as well. As obvious, deepening a particular normative consideration in any machine learning task is expected to introduce constraints that would reduce the clustering quality overall; in other words, higher local connectivity is not expected to come *'for free'*. A good clustering under the local connectivity lens would be one that can deepen local connectivity with *limited impact* on other metrics of clustering quality; this, we will see, is the focus of our empirical evaluation.

5.2 LOFKM: The Method

In line with the idea of bringing cluster prototypes closer to higher LOF data objects, we start with assigning a weight to each data object, as follows:

$$W(X) = \begin{cases} 1.0 & LOF(X) \leq 1 \\ LOF(X) & otherwise \end{cases} \tag{10}$$

$W(X)$ is simply the LOF score bounded under by 1.0. This weight is then used in re-formulating the standard K-Means objective as follows, for a given clustering $\mathcal{C}$ over the dataset:

$$\sum_{C \in \mathcal{C}} \sum_{X \in C} W(X) \times \left(\sum_{A \in \mathcal{A}} (X.A - C.A)^2 \right) \tag{11}$$

where A is any attribute from the set of attributes $\mathcal{A}$, with $X.A$ and $C.A$ denoting the value taken for the attribute by the object X and the cluster prototype of cluster C respectively (notice that we have overloaded C to denote both the cluster and its prototype for notational simplicity). Intuitively, this is equivalent to considering the dataset as comprising each object as being replicated as many times as its LOF score requires, and applying standard K-Means over the enlarged dataset. There are two sets of variables that we can change in order to optimize for the objective; the *cluster memberships* and *cluster prototypes*. Standard K-Means optimizes these in turn (keeping one set fixed, and optimizing for the other) over many iterations until the cluster memberships stabilize.

[5] https://en.wikipedia.org/wiki/Star_and_crescent

Name	# Instances	# Attributes	# Classes
Yeast	1484	8	10
Wireless[6]	2000	7	4
Avila	20867	10	12

Table 1: Dataset Statistics

Under the objective in Eq 11, the membership assignment step, given the cluster prototypes, is as follows:

$$\forall\, X \in \mathcal{X},\;\; Cluster(X) = \arg\min_{C \in \mathcal{C}} \sum_{A \in \mathcal{A}} (X.A - C.A)^2 \qquad (12)$$

Since we are updating each object independently given the current estimate of cluster prototypes, $W(X)$ does not factor into this cluster assignment step since it is simply a constant factor for each X independent of which cluster X gets assigned to. This, as one may notice, is *exactly the cluster assignment step in K-Means*. It may sound odd as to why we critique the K-Means membership desert and still use it in *LOFKM*; the crucial factor here is that this proximity-based membership desert is used against a set of cluster prototypes that are estimated in very sharp contrast to the analogous step in K-Means. The *LOFKM* cluster prototype estimation step is as follows:

$$\forall\, A \in \mathcal{A},\;\; C.A = \frac{\sum_{X \in C} W(X) \times X.A}{\sum_{X \in C} W(X)} \qquad (13)$$

In other words, each X is accounted for as many times as warranted by $W(X)$.

Towards generating a clustering from a dataset, much like in K-Means clustering, we start with a random initialization of cluster prototypes followed by iteratively applying Eq 12 and Eq 13 until the cluster memberships become relatively stationary across iterations. Owing to these steps mirroring those of standard K-Means, we do not outline a full pseudocode for *LOFKM* herewith.

5.2.1 Note on Complexity. The K-Means steps, much like the usual K-Means algorithm, is linear in the number of objects, number of clusters and number of attributes. However, computing the weights, i.e., Eq 10, is more expensive. While LOF computation is generally regarded as between superlinear and quadratic in the number of objects [7], faster methods have recently been proposed [3, 14]. It is notable that any further advancements in improving LOF computations readily transfer over to LOFKM as well, given that the LOF and K-Means steps are decoupled within LOFKM.

6 EXPERIMENTAL EVALUATION

We now describe our empirical evaluation. We start by outlining the datasets and baselines in our empirical evaluation, while also outlining the evaluation setup. This is followed by detailed results from empirical evaluation and analyses.

6.1 Datasets, Baselines and Evaluation Setup

6.1.1 Datasets. We evaluate our methods on multiple real-world datasets from the UCI Machine Learning Repository. These have widely different numbers of objects, ranging from $1.5k$ to $21k$, and spread across $4-12$ classes. The dataset statistics are summarized in Table 1.

6.1.2 Baseline. Much like the only two existing papers that propose new normative considerations in clustering, that of proportionality [8] and representativity [19], we use the classical K-Means formulation as the baseline method in our experimental evaluation. We do not include either of the above methods in our comparison since they optimize for significantly different notions of membership desert; as an example, it may be seen that the method from [8] was used in the empirical evaluation for representativity in [19], and it was observed (unsurprisingly) that the basic K-Means fared much better than [8] on representativity.

6.1.3 Evaluation Setup. We follow the evaluation framework for fair clustering (as in [1, 19]), with the evaluation being conducted across two kinds of metrics; (i) *local connectivity* (analogous to fairness metrics in fair clustering) metrics, viz., *AvgLCD* and *MaxLCD*, and (ii) clustering quality metrics, viz., *silhoutte [21]* (Sil) and *clustering purity*[7] (Pur). For *LOFKM*, we expect improvements on the former, and setbacks on the latter. *LOFKM* may be judged to be effective if it is able to achieve good gains on the former set of metrics, at reasonable detriment to the latter. For both LOFKM and K-Means, we average the performance metrics across 100 random starts, so as to achieve stable and reliable numbers. We always set the number of clusters in the output, i.e., the parameter K, to be equal to the number of classes in the respective datasets (Ref. Table 1).

6.2 Experimental Results and Analysis

We first outline the structure of the experimental analysis. Local connectivity, as outlined in Sec 4.2, is assessed using a parameter t, the number of relevant neighbors for an object; this parameter is used in the computation of both *MaxLCD* and *AvgLCD*. For *LOFKM*, there is a similar parameter in the input, which is the number of neighbors for an object used in LOF computation (Ref. Sec. 2.2). These, being similar in spirit, are set to identical numbers, and we denote both as t. We experiment with varying values of t; in the interest of brevity, we report results for $t \in \{3, 4, 5\}$ as a representative set of results since the trends held good for higher values. *KM*, short for K-Means, does not use any neighborhood parameter in the method.

The evaluation on fairness metrics is illustrated in Table 2 whereas the evaluation on clustering quality appears in Table 3. The percentage change of the *LOFKM* metric over that in *KM* is indicated explicitly, for ease of interpretation. An average of $5-10\%$ gains are achieved on the *AvgLCD* measure, indicating a sizeable improvement in local connectivity in the clusterings output by *LOFKM* over those of *KM*. Further, the improvements are seen to improve with the size of the dataset, which is expected since larger datasets allow for more flexibility in clustering assignments. The corresponding improvements in *MaxLCD* are seen to be smaller. *MaxLCD* quantifies the worst local connectivity across clusters, and thus relates to the quantification over a single cluster, which in turn is the worst local connectivity across members of the cluster. While it would intuitively be expected that least locally connected objects which

[7] https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

Dataset	Method	*AvgLCD* ↓			*MaxLCD* ↓		
		t = 3	*t = 4*	*t = 5*	*t = 3*	*t = 4*	*t = 5*
Yeast	KM	0.93	1.18	1.42	1.20	1.57	2.00
	LOFKM	0.92	1.08	1.15	1.15	1.53	1.98
	Improvement %	01.07%	08.47%	09.01%	04.17%	02.55%	01.00%
Wireless	KM	1.24	1.68	2.04	1.32	1.79	2.24
	LOFKM	1.18	1.56	1.90	1.31	1.73	1.95
	Improvement %	04.83%	07.14%	06.87%	00.76%	03.35%	12.95%
Avila	KM	1.11	1.48	1.83	1.33	1.77	2.19
	LOFKM	0.99	1.31	1.61	1.32	1.80	2.19
	Improvement %	10.81%	11.49%	12.02%	00.75%	-01.69%	00.00%
Avg of Improvement %		05.57%	09.03%	09.30%	01.89%	01.40%	04.65%

Table 2: Evaluation on Local Connectivity Measures. Note that lower values are better for both *AvgLCD* and *MaxLCD*, as indicated using the arrow in the column heading.

Dataset	Method	*Sil* ↑			*Pur* ↑		
		t = 3	*t = 4*	*t = 5*	*t = 3*	*t = 4*	*t = 5*
Yeast	KM	0.26			0.42		
	LOFKM	0.27	0.27	0.26	0.41	0.41	0.41
	Change %	+03.84%	+03.84%	00.00%	-02.40%	-02.40%	-02.40%
Wireless	KM	0.40			0.93		
	LOFKM	0.39	0.39	0.39	0.77	0.78	0.78
	Change %	-02.50%	-02.50%	-02.50%	-17.20%	-16.13%	-16.13%
Avila	KM	0.15			0.46		
	LOFKM	0.18	0.18	0.18	0.45	0.45	0.45
	Change %	20.00%	20.00%	20.00%	-02.17%	-02.17%	-02.17%
Avg of Change %		07.11%	07.11%	05.83%	-07.26%	-06.90%	-06.90%

Table 3: Evaluation on Clustering Quality Measures. Note that higher values are better for both *Sil* and *Pur*, as indicated using the arrow in the column heading.

would be in sparse regions where local connectivity improvements would be harder to achieve, it is promising to note that *LOFKM* consistently achieves improvements on *MaxLCD* over *Yeast* and *Wireless*; the corresponding improvements in *Avila* are limited, and negative in one case. The trends on the clustering quality metrics in Table 3 may be regarded as quite interesting. It may be noted that t does not play a role for results of *KM* since the clustering quality metrics as well as *KM* are agnostic to t. As outlined earlier, we expect that the cost of local connectivity enhancement in *LOFKM* would manifest as a deterioration in clustering quality. While we can observe such deterioration in *Pur* in Table 3, *LOFKM* is surprisingly able to achieve improvements in *Sil* on the *Yeast* and *Avila* datasets. On careful investigation, we found evidence to hypothesize that *LOFKM* discovers *secondary clustering structures*, which differ from the primary ones that are better correlated with external labels (*Pur*, as one might remember, measures correlation with external labels). These secondary clustering structures, while not necessarily tighter, are found to be well separated, yielding improvements in *Sil*. This interestingly correlates with similar observations over *Sil* in representativity fairness (Ref. Sec. 6.3.2 in [19]). In contrast to *Yeast* and *Avila*, *Wireless* does not seem to exhibit such well-separated secondary structures, leading to falls in both *Sil* and *Pur*. Across datasets, the deterioration in *Pur* is seen to be fairly limited, to within 10%; we would re-iterate that the fact that such deterioration comes with an improvement in *Sil* indicates the promisingness of *LOFKM*. To summarize, *LOFKM* is seen to offer consistent and often sizeable improvements in local connectivity, with mixed trends in clustering quality.

7 DISCUSSION

Having considered local connectivity as a factor for membership desert in clustering, it is useful to think about how this relates to other notions and other factors that may be argued to play a role in membership desert.

Local connectivity is distinctly different from *representativity* [19] in that an object that is very distant from the cluster prototype could still be locally connected to the very same cluster. While this conceptual distinction cannot be more apparent, in practice, we expect peripheral/fringe objects of a cluster to suffer from local connectivity, and similar could be true for representativity as well. In a way, local connectivity provides a way to distinguish between objects in the periphery of clusters that are locally connected to the cluster and those that are not. This points to the possibility of using both in tandem. Peripheral objects 'deserve' better representativity, but

local connectivity could provide a way to prioritize among them. The connection with *proportionality* [8] is somewhat more nuanced, since proportionality violations are evaluated at the collection level. That said, proportionality violations may be expected to be in the gulf between existing clusters, since those would be the locations where one would expect to see preference to the existing cluster assignment waning. Thus, addressing proportionality violations by changing cluster assignments may be seen as automatically addressing local connectivity, since the objects would be better locally connected to the new cluster. These relationships between concepts could lead to interesting future explorations.

Membership desert having been considered along lines of *proximity* [19], *collective vote* [8] and *local connectivity*, it is interesting to think of whether there are other ways of thinking about cluster memberships. The building blocks of Silhoutte [21] provide an interesting angle to the issue. Silhoutte quantifies the average distance to the objects of it's existing cluster, and those to the objects of the *next nearest* cluster, and uses these to compute a normalized difference, called the object-specific silhoutte co-efficient. The silhoutte score is then the mean[8] of these. It may be argued that each object needs to be accorded a minimum level of higher proximity to the existing cluster than the next best, or that objects need to score similarly on their respective silhoutte co-efficients. This line of exploration requires low variance of the silhoutte co-efficients over the dataset, as well as maximizing the minimum silhoutte co-efficient. Another perspective is to consider the role of sensitive attributes such as race, sex, gender and religion, when clustering person-level data. Each of the notions of membership desert could be extended using the role of sensitive attributes. For example, there could be two routes to enhance membership desert based on the relationship with the cluster prototype. One could be through proximity, and another could be through similarity in sensitive attribute profiles, and these could compensate slightly for each other. This discussion hopefully serves to indicate that there is plentiful meaningful room for enhancing the diversity of membership desert notions in clustering formulation. A recent position paper [18] considers certain other normative possibilities within the task of clustering.

8 CONCLUSIONS AND FUTURE WORK

In this paper, we investigated, for the first time, local connectivity and its relevance to membership desert in centroid clustering. Through a critique of cluster membership desert focusing on K-Means, we motivated the need to consider local connectivity as a crucial normative consideration in deciding cluster memberships, in addition to centroid proximity (the only criterion in classical formulations such as K-Means). Following upon this argument, we outlined ways of quantifying local connectivity for a given clustering, to aid evaluating clusterings on the local connectivity criterion. Towards developing a clustering that would promote local connectivity, We considered local neighborhood assessments from the family of density-based clustering methods, and adopted *LOF* for usage within the K-Means formulation, leading to a local connectivity-oriented clustering method, *LOFKM*. Through an evaluation of *LOFKM* vis-a-vis K-Means (following the evaluation frameworks in similar works [8, 19]), we illustrated that *LOFKM* is able to deepen local connectivity in clustering outputs while producing well-separated clusters at only reasonably degradations to clustering purity as measured against external labels.

Future Work: We are considering various layers of interplay between local connectivity and notions of fairness as espoused within popular schools such as *Rawlsian fairness* [13]. Second, we are considering blending local connectivity along with the other normative principles explored in clustering, such as representativity and proportionality. Third, we are considering other criteria for membership desert involving sensitive attribute classes such as gender and ethnicity. Further, we have also been considering the relationships between clustering interpretability (e.g., [4]) and fairness.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

REFERENCES

[1] Savitha Abraham, P Deepak, and Sowmya Sundaram. 2020. Fairness in Clustering with Multiple Sensitive Attributes. In *EDBT*.

[2] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

[3] Kasra Babaei, ZhiYuan Chen, and Tomas Maul. 2019. Detecting Point Outliers Using Prune-based Outlier Factor (PLOF). *arXiv preprint arXiv:1911.01654* (2019).

[4] Vipin Balachandran, P Deepak, and Deepak Khemani. 2012. Interpretable and reconfigurable clustering of document datasets by deriving word-based rules. *Knowledge and information systems* 32, 3 (2012), 475–503.

[5] Suman Bera, Deeparnab Chakrabarty, Nicolas Flores, and Maryam Negahbani. 2019. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems*. 4955–4966.

[6] Reuben Binns. 2020. On the apparent conflict between individual and group fairness. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 514–524.

[7] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[8] Xingyu Chen, Brandon Fain, Charles Lyu, and Kamesh Munagala. 2019. Proportionally Fair Clustering. In *ICML*.

[9] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, and Sergei Vassilvitskii. 2017. Fair clustering through fairlets. In *NIPS*. 5029–5037.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.

[11] Anil K Jain. 2010. Data clustering: 50 years beyond K-means. *Pattern recognition letters* 31, 8 (2010), 651–666.

[12] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.

[13] Rawls John. 1971. A theory of justice. (1971).

[14] Jihwan Lee and Nam-Wook Cho. 2016. Fast outlier detection using a grid-based algorithm. *PloS one* 11, 11 (2016).

[15] Michele Loi and Markus Christen. 2019. How to Include Ethics in Machine Learning Research. *ERCIM News* 116, 3 (2019).

[16] James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Vol. 1. Oakland, CA, USA, 281–297.

[17] Fionn Murtagh. 1983. A survey of recent advances in hierarchical clustering algorithms. *The computer journal* 26, 4 (1983), 354–359.

[18] Deepak P. 2020. Whither Fair Clustering?. In *AI for Social Good Workshop*. Harvard CRCS.

[19] Deepak P and Savitha Sam Abraham. 2020. Representativity Fairness in Clustering. In *ACM Web Science*.

[20] Leonard KAUFMAN Peter J RDUSSEEUN. 1987. Clustering by means of medoids. (1987).

[21] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[22] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)* 42, 3 (2017), 1–21.

[23] Éric D Taillard. 2003. Heuristic methods for large centroid clustering problems. *Journal of heuristics* 9, 1 (2003), 51–73.

Implementation of Dynamic Page Generation for Stream Data by SuperSQL

Keita Terui
Keio University
Yokohama, Japan
terui@db.ics.keio.ac.jp

Kento Goto
Keio University
Yokohama, Japan
goto@db.ics.keio.ac.jp

Motomichi Toyama
Keio University
Yokohama, Japan
toyama@ics.keio.ac.jp

ABSTRACT

SuperSQL is an extension of SQL that allows you to structure the output of relational databases by writing your own queries and to express various layouts. However, this method is not suitable for data with high update frequency, such as stream data, because the information in the database refers to the data at the time of SuperSQL execution. In this study, we propose an implementation of a web page generation function that asynchronously updates a web page with the latest information for frequently updated data, using PipelineDB and SuperSQL, both of which are DBMSs capable of processing streams. You can specify the dynamic part of the stream by specifying the stream in the "decorator" which is a feature of SuperSQL. At the same time, you can specify "pull" and "push" in the stream decorator to select how the dynamic part is updated. This makes it possible to create a web page that displays the latest stock prices at any time in a page that displays a list of stock prices.

CCS CONCEPTS

• **Information systems → Hypertext languages**.

KEYWORDS

IDEAS, International Database Engineering & Applications Symposium

ACM Reference Format:
Keita Terui, Kento Goto, and Motomichi Toyama. 2020. Implementation of Dynamic Page Generation for Stream Data by SuperSQL. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410566.3410607

1 INTRODUCTION

Syntactically, SuperSQL is an extension of SQL that allows you to structure the output of relational databases by writing your own queries and to express various layouts. Although ordinary SQL can produce arbitrary flat tables, SuperSQL can create various tables including nesting, in three or more dimensions. In addition, by using SuperSQL, it is possible to generate web pages, php, etc. without coding, so it can be regarded as a so-called no-code tool.

In current implementation of SuperSQL, all attribute data described in a query is retrieved from backend SQL DMBS at once to generate web pages. Therefore, the data is referred to the database information at the time of SuperSQL execution, and it is difficult to handle thedata with high update frequency such as stream data. The purpose of this study is to revise the HTML generator in SuperSQL, which updates pages asynchronously with the latest information on frequently updated data, and to generate more effective web pages by adopting PipelineDB, a DBMS that supports stream processing. The only syntactical extension to SuperSQL language is the addition of 'stream' decorator, which specifies a portion of web page to be updated asynchronously. We have implemented the asynchronous web page generation by two methods, one using Ajax and the other using Server Sent Events. When users generate asynchronous web pages using SuperSQL, we propose a function that allows them to choose the method of data acquisition between the browser's technical support and real-time performance.

In this paper, Chapter 2 describes SuperSQL, Chapter 3 describes related technologies and research, Chapter 4 describes the generation of dynamic web pages based on stream data by SuperSQL, Chapter 5 describes experiments and evaluations, and Chapter 6 describes a summary.

2 RELATED TECHNOLOGY AND RELATED WORK

2.1 PipelineDB

PipelineDB[5] is an open source streaming SQL relational database. It is compatible with PostgreSQL and can query the stream data with PostgreSQL-compatible queries. By flowing a large amount of data on the system onto PipelineDB, we can continuously execute SQL queries for a certain period of time. This query is called a continuous query, and in a system that handles static data such as PostgreSQL, it stores the data and returns the result only once when the query is issued. PipelineDB, on the other hand, prepares queries in advance and returns results each time the data flows. The part where the data flows is a table-like declaration method called Stream, which can be used to flow data to the Stream with an INSERT statement. Stream does not keep the streamed data itself, but keeps only the aggregate results of the streamed data through a mechanism called Continuous View, which is similar to SQL view, and does not need much space as a database. It is possible to simplify the ETL work such as data extraction, formatting, and introduction, because it is possible to complete the process from the introduction to the aggregation of stream data with SQL queries.

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06.
https://doi.org/10.1145/3410566.3410607

Although Apache Kafka[6], Spark Streaming[7], and Amazon Kinesis[8] are used as stream data processing platforms, PipelineDB is used in this paper because PostgreSQL has been used as a typical DBMS in SuperSQL for a long time and there is already an infrastructure to connect to PostgreSQL.

2.2 Data Acquisition Through Asynchronous Communication

The asynchronous communication in this paper is realized on the basis of Ajax, which performs asynchronous communication using JavaScript and DOM, which has attracted attention since the appearance of the google map, and Server Sent Events, which is a kind of html5-related API proposed at W3C.

As a way to update data asynchronously using Ajax, we check the server at a configurable interval for the latest data from a client called polling. The frequency of the pulls must be high to guarantee high accuracy data, but too many pulls can lead to redundant checks and increase network traffic. On the other hand, less frequent pulls may result in missed updates.

One way to update data asynchronously using Server Sent Events is to transfer the data from the server to the client by means of an HTTP server push. In this case, the server does not terminate the connection after returning the response data to the client. This feature allows the client to receive data sent intermittently from the server at any time, without the cost of reconnecting the client and server.

3 SUPERSQL

In this chapter, we briefly describe SuperSQL. The architecture of SuperSQL is shown in Figure 1.

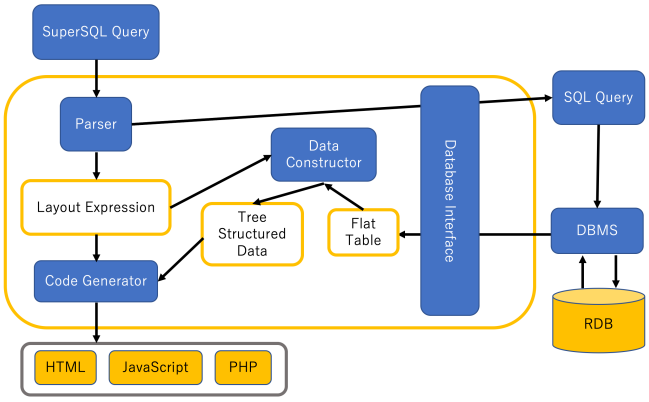


Figure 1: SuperSQL Architecture

SuperSQL is an extension of SQL that enables to structure the output of relational databases and to express various layouts. It has been developed in Toyama Laboratory, Keio University[1, 2]. The query is a SQL SELECT clause replaced by a GENERATE clause with the syntax GENERATE <*media*> <*TFE*>. TFE[3] is Target Form Expression. TFE is an extension of the target list of the SQL. The common attribute of a target list are separated by commas, but the structure of the output document is more complex than a list separated by commas. TFE therefore introduces several operators, the horizontal operator (fist dimension), the vertical operator (second dimension), and the depth operator (third dimension). Some special features are also available. <*TFE*>. Here, <*media*> indicates the output medium, which can be HTML, PDF, Mobile_HTML5[4], etc. <*TFE*> stands for Target Form Expression, which is an extension of the target list, and is a kind of expression with layout specification operators such as concatenators and iterators.

3.1 Connector

A joiner is an operator that specifies the direction (dimension) in which the data obtained from the database is combined, and there are three types of joiners as follows. The parentheses show the operators in the query.

- Horizontal Connector(,) Combines data horizontally and outputs it.
  Example. name, place

name	place

- Vertical connector(!) Combines data horizontally and outputs it.
  Example. name! place

name
place

3.2 Repeater

The repeater in the specified direction, as many times as there are values in the database. An iterator does not only specify a structure, but can also specify the relationship between attributes by their nesting relationships. For example,

[Deployment]!, [Employer]!, [Salary]!

then there is no relationship between the attributes, just a list of each. On the other hand, using the nest,

[Deployment! [Employer, Salary]!]!

the relationship between the attributes is specified, for example, a list of the names of employers and salaries for each department is displayed. The types are described below.

- Horizontal repeater([ ],) As long as there is a data instance, the data for that attribute is repeated horizontally.
  Example. [Name],

name1	name2	...	name10

- Vertical repeater([ ]!) Displays the data of the attribute vertically as long as there is a data instance.
  Example. [Name]!

name1
name2
...
name10

3.3 Decorater

SuperSQL can add information such as character size, character style, width, character color, background, height, and position to information extracted from related databases. These are specified

by the Decorater (@)

<Attribute>@{<Decorater>}

Decorater is specified as “name of Decorater = its content”. To specify more than one, each of them is separated by “,”.

Example. [name@{width=100, color=red}]!

4 DYNAMIC PAGE GENERATION WITH SUPERSQL

Here, we describe the generation of a dynamic Web page corresponding to the stream data by our proposed method.

4.1 Specifying the Asynchronous Update Portion by Query

The proposed method generates Web pages that are updated asynchronously at the specified time by specifying the Decorater @{stream=n} to the SuperSQL iterator. If n is not specified, the update is performed in 1000 milliseconds. n is a number, meaning that the update is performed in n milliseconds. The following declarations are implemented as Decorater.

Declaring a stream

```
[ TEF1, [ TEF2 ]! ]!@{stream=n1},
[ TEF1, [ TEF2, [TEF3] ]! ]!@{stream=n2}
```

Declaring a stream with pull/push

```
[ TEF ],@{stream=n, mode='pull'},
[ TEF ],@{stream=n, mode='push'}
```

4.2 Architecture

The architecture of the proposed method is shown in Figure 2. The user generates HTML, CSS, JavaScript, and PHP by executing a SuperSQL query.

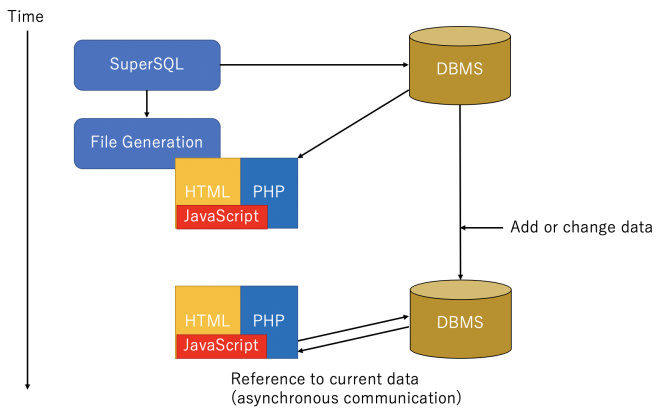


Figure 2: Architecture of the proposed method

By executing PHP, data is queried in the database, and JSON format data is generated. The viewer is shown the HTML and JavaScript data received.

5 IMPACT OF IDEAS

5.1 Data Acquisition Mechanism

In the proposed method, there are two data acquisition methods as described in 2.2. In this paper, we describe two types of data acquisition methods proposed in this paper.

Table 1: Data acquisition method

	pull	push
Technology	Ajax	Server Sent Events
Cost	High	Low
Browser compatibility	High	Low
Real time	Low	High

Table 1 outlines the two types of data acquisition methods. pull mode can be specified by writing “mode=‘pull’ ” with the ornament stream, and even if no mode is specified, pull mode is used. pull mode is characterized by high communication cost, low real-time performance, high browser compatibility, and low need to consider the browser type and version of the viewer. Figure 3 shows the architecture in the pull mode.

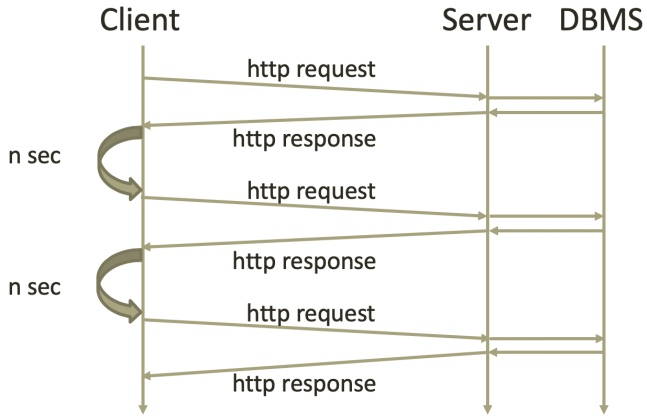


Figure 3: Architecture in pull mode

In push mode, once a client-server connection is established using Server Sent Events, the server can send data to the client unilaterally, but without request, and the client can send data to the server in the push mode by writing ‘mode=‘push’ ” with the ornament stream. While the cost of communication is low because there is no need to re-establish the connection, and it is highly real-time, the problem is that it does not work with older browsers. Figure 4 shows the architecture in the push mode.

For the purpose of explanation, a database for a user who wants to monitor the rate of a virtual currency and a specific virtual

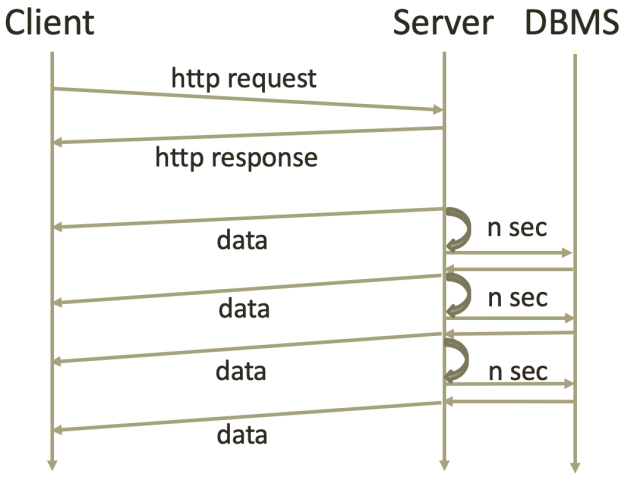


Figure 4: Architecture in pull mode

currency is given here as an example. The database used in the example is as follows.

- crypto(id, exchange, name, rate)
- users(id, name)
- monitor(c_id, u_id)

"crypto" is the Continuos View described in 2.1 and is the rate of the virtual currency handled by a given exchange against the current Japanese Yen. "users" is a table of people who monitor the rate of a given virtual currency; monitors is an intermediate table representing the relationship between crypto and users; and users is an intermediate table representing the relationship between crypto and users. The Stream, the crypto reference, is constantly INSERT the rate for each virtual currency.

sample1.ssql

```
GENERATE HTML_stream
[ null((asc)u.id), u.name,
[ c.exchange, c.name, c.rate
]!
]!@{stream=3000}
FROM users u, crypto c, monitor m
WHERE c.id = m.c_id AND u.id = m.u_id
```

By executing sample1.sql, we get the results shown in Figure 5.

In Figure 5, the area enclosed by the blue frame is updated asynchronously.

User1	Zaif	BCH	17005
	Zaif	BTC	433050
User2	bitflyer	BTC	435001.4
User3	Zaif	BTC	433050
	Zaif	MONA	134
	bitflyer	BTC	435001.4
User4	bitflyer	BTC	435001.4
User5	Zaif	BCH	17005
	Zaif	BTC	433050
	Zaif	MONA	134
	Zaif	XEM	6.8601
	bitflyer	BTC	435001.4
	bitflyer	FX_BTC	435084.9

Figure 5: The result of executing sample1.sql

sample2.sql

```
GENERATE HTML_stream
[image(c.exchange, './picts')],!
[(asc)c_zaif.name, c_zaif.rate]!@{stream=3000},
[(asc)c_bitflyer.name, c_bitflyer.rate]!@{stream=1000}
FROM crypto c, crypto c_zaif, crypto c_bitflyer
WHERE c_zaif.exchange='Zaif'
OR c_bitflyer.exchange='bitflyer'
```

By executing sample2.sql, we get the results shown in Figure 5.

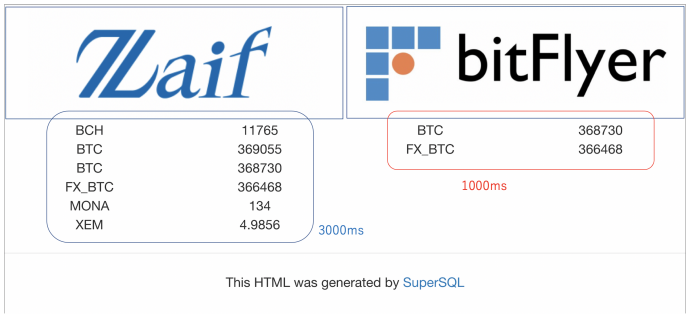


Figure 6: The result of executing sample2.sql

In Figure 6, multiple stream Decoraters are used.

6 EXPERIMENTS AND EVALUATIONS

The following evaluation experiments were conducted to evaluate the usefulness of the proposed system.

- Client load test by update frequency
- Load tests on PipelineDB on the server

6.1 Client Load Test by Update Frequency

In this experiment, we conducted a load experiment by the asynchronous update time of a web page and the CPU usage of the client side. The experimental environment of the server is as follows.

- Server
- OS: CentOS 7.3.1611
- CPU: Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz 10x2core
- Memory: 126GB
- DBMS: PipelineDB 0.9.9

The experimental environment of the client is as follows.

- Client
- OS: Mac OS Mojave
- CPU: 3.1 GHz Intel Core i5
- Memory: 16GB
- Browser: firefox 64.0

In this experiment, using sample1.sql to specify the pull mode or push mode?, and we measured the CPU usage of a client browsing a product with the frequency of web page refreshes set to 100, 200, ... In this experiment, we measured the CPU usage of clients browsing the product with the refresh frequency set to 100, 200, ... , 1000 ms. Figure 7 shows the experimental results of the client load experiment with the update frequency.

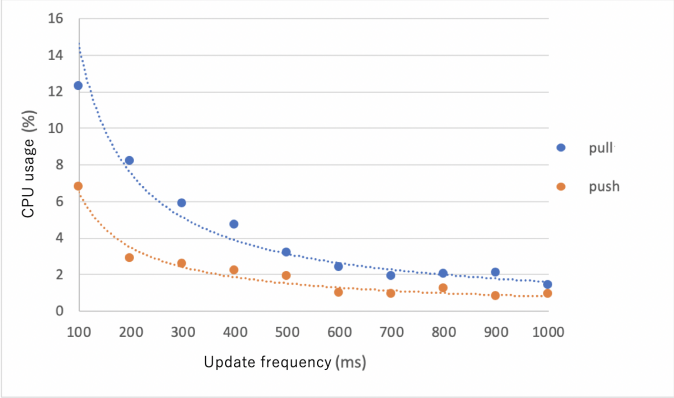


Figure 7: Experimental results of the client load experiment by update frequency

The results show that on the client side, pull mode consumes more CPU resources than push mode. On average, the pull mode consumed 2.1 times more CPU resources than the pull mode. This is thought to be because in pull mode, the browser sends a request to the server for updating and displays the data from the received data, whereas in push mode, the browser does not need to send a request to the server and only displays the data sent to it.

6.2 Load tests on PipelineDB on the server

In this experiment, using sample1.sql to specify the pull mode, and we calculated the number of clients who can access the same web page from the CPU usage of PipelineDB on the server. The experimental environment of the server and client is the same as in 6.1.We measure the CPU usage of PipelineDB on the server when 1, 2, ..., 10 simultaneous requests are sent to a web page that is updated asynchronously in 10 milliseconds. Figure 8 shows the experimental results of a load experiment on PipelineDB on the server.

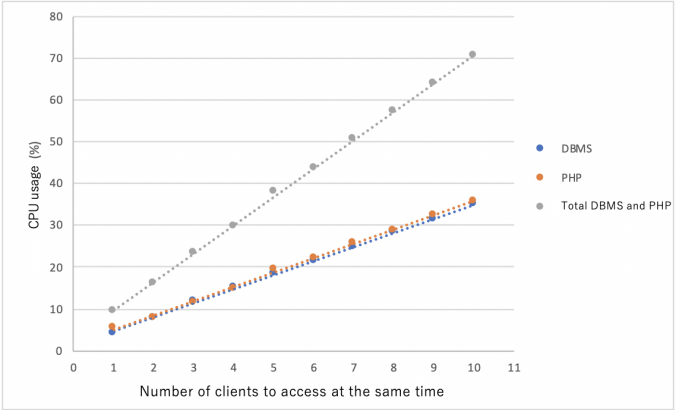


Figure 8: Load Tests on PipelineDB on The Server

The results show that there is a proportional relationship between the number of concurrent accesses and the CPU usage of PipelineDB on the server. From this relationship we can see the following

$$(\text{CPU utilization}) \propto (\text{Number of simultaneous accesses}) \quad (1)$$

In addition, the number of simultaneous accesses in one second from the server can be expressed as follows.

$$(\#\text{ simultaneous accesses}) = \frac{(\#\text{ clients}) \cdot (\#\text{ PHP files}) \cdot (1\text{ s})}{(\text{Update Interval})} \quad (2)$$

Therefore, the relationship between the maximum number of clients who can access the product of this research and the CPU resources allocated to a web page can be expressed as follows

$$(\text{Maximum number of clients}) \propto \frac{(\text{CPU usage}) \cdot (\text{Update interval})}{(\text{Number of PHP files}) \cdot (1\text{ s})} \quad (3)$$

Considering the thread of the server machine, it can be expressed as follows.

$$(\text{Maximum \# clients}) \propto \frac{(\text{CPU usage}) \cdot (\#\text{ Threads}) \cdot (\text{Update interval})}{(\#\text{ PHP files}) \cdot (1\text{ s})} \quad (4)$$

7 CONCLUSION.

In this paper, we implement a feature to generate dynamic web pages corresponding to stream data using SuperSQL. This makes it possible to create asynchronous web pages using only SuperSQL queries, instead of having to write multiple files, such as HTML, JavaScript, and PHP, respectively, to create asynchronous web pages.

REFERENCES

[1] SuperSQL: http://ssql.db.ics.keio.ac.jp/
[2] M. Toyama: "SuperSQL: An Extended SQL for Database Publishing and Presentation", Proceedings of ACM SIGMOD'98 International Conference on Management of Data, pp.584-586, 1998.

[3] Toshiyuki Seto, Takuhiro Nagafuji and Motomichi Toyama. Generating HTML Sources with TFE Enhanced SQL, In *ACM Symposium on Applied Computing*, pp.96-100, 1997.
[4] K. Goto and M. Toyama, "Mobile Web Application Generation Features For SuperSQL", in Proceedings of the 20th International Database Engineering & Applications Symposium, IDEAS 2016, pp. 308-315, 2016.
[5] PipelineDB: https://www.pipelinedb.com/
[6] Jay Kreps, Neha Narkhede, and Jun Rao. Kafka: a distributed messaging system for log processing. ACM SIGMOD Workshop on Networking Meets Databases. page 6. 2011.
[7] Spark Streaming: https://spark.apache.org/streaming/
[8] Amazon Kinesis: http://aws.amazon.com/kinesis/

HYRAQ: Optimizing Large-Scale Analytical Queries through Dynamic Hypergraphs

Mustapha Chaba Mouna
LRDSI Laboratory, Faculty of Science
University Blida 1
Blida, Algeria
mustapha.medea@gmail.com

Ladjel Bellatreche
LIAS/ISAE-ENSMA
Futuroscope, France
bellatreche@ensma.fr

Narhimene Boustia
LRDSI Laboratory, Faculty of Science
University Blida 1
Blida, Algeria
nboustia@gmail.com

ABSTRACT

In critical situations, making quick and precise decisions requires a rapid execution of a large amount of concurrent navigational and exploratory queries over collected data stored in repositories such as data warehouses. To satisfy the decision-maker's requirement, a deep understanding of the properties of these queries is necessary. In addition to their large-scale, they are ad-hoc, dynamic and highly interacted. By a quick analysis of these properties, we figure out that the first three are factual whereas the last one is behavioral. The literature has widely reported that the interaction of analytical queries has a crucial impact on selecting optimization structures (e.g., materialized views) in data storage systems. By keeping these four properties in mind, it becomes a necessity to find scalable and efficient data structures to simultaneously model them for better optimization of large-scale queries. In this paper, we first show the crucial role of the interaction phenomenon in optimizing concurrent data and mining queries by identifying its limited capacity in considering all factual properties. Secondly, we propose a dynamic hypergraph as a data structure to manage the four above properties and we show its great contribution in selecting materialized views. Finally, intensive experiments are conducted to evaluate the efficiency of our proposal and its connectivity with a commercial DBMS.

CCS CONCEPTS

• **Information systems** Query optimization;

KEYWORDS

Query Interaction, Query Volume, Dynamic Hypergraphs & Views Selection

ACM Reference Format:
Mustapha Chaba Mouna, Ladjel Bellatreche, and Narhimene Boustia. 2020. HYRAQ: Optimizing Large-Scale Analytical Queries through Dynamic Hypergraphs. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410582

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410582

1 INTRODUCTION

The covid-19 virus highlights several Information technologies such as Data Warehouses, Big Data, Linked Open Data, Machine Learning, Data Science, Robotics, Image processing, Drones, Advanced storage infrastructures, Chatbots, etc. to track and fight this pandemic. Big companies such as Alibaba, Baidu, Huawei and startups (e.g., BlueDot[1]) are working tirelessly to provide efficient and quick solutions. For instance, the Health Code system developed by the Chinese government is one of the examples of Big Data usage. It allows identifying and assessing the risk of each individual based on their *travel history*, how much time they have spent in virus hotspots, and potential exposure to people carrying the virus.

All these provided services are data-enabled. Therefore, having the *right data* at the *right time* has become an imperative issue in making relevant and saving decisions. If we stay in the context of Covid-19, we realize that several organizations *collect* data, *displays* it and *makes* it freely available through a GitHub - the case of the Johns Hopkins University, USA [2] or as open data repositories such as the Genbank[3]. These repositories are accessed by research teams around the world to feed their work using navigational and exploratory large scale queries [3, 13, 27]. Surprisingly, analyzing some of these repositories, we figure out that they can be easily modeled by the data warehouse (DW) technology. One of the main important characteristics of these repositories is their ability to manage large-scale queries. This situation has been identified (with a certain degree of similarity) in the SQLShare system - a multi-year SQL-as-a-Service Experiment that can manage 11 137 SQL statements [16]. These queries have to be quickly optimized to speed up the decision-maker processes. In addition to their volume, they are also known by the other three properties: ad-hoc, dynamic and share similar operations. These 4-properties do not get the same attention by the research community, where they were partially considered.

The query operation sharing is a guiding feature and at the same time impacted by the three others. The identification of common subexpressions of queries is the key issue for the performance of multi-query processing [12]. Historically, the Problem of Multi-Query Optimization ($PMQO$) has been largely studied in 80's [11, 24] and then in all database generations without any exception. The $PMQO$ has largely contributed to solving instances of the physical design problem such as materialized views selection [30], data partitioning schemes selection [2], known as NP-hard problem. It has also contributed to resolving concurrent mining queries [29], and recently in Big Data Warehouses [9]. In all database generations

[1] https://bluedot.global/
[2] https://github.com/CSSEGISandData/COVID-19
[3] https://www.ncbi.nlm.nih.gov/genbank/

in which the physical design is important such as DW, the $PMQO$ and the instances of the physical design problem are usually tackled in an isolated way, despite their strong dependency. The $PMQO$ has usually been studied for a static and a priori known workload, where several variants of the A* heuristic have been proposed for finding optimal solutions to the moderately sized (up to ten queries) [26]. Note that the materialized views selection problem (VSP) considered one of the most studied problems in DWs has been studied in static and dynamic contexts, without really taking into account its interaction with $PMQO$, except the work of [30] that uses a unified graph (called Views Processing Plan) that merges all local query trees. This study is considered as the pioneer in the context of DW that deals with the process of constructing a unified query graph dedicated to the process of selecting materialized views. The main drawback of this work concerns the scalability issue [7]. To overcome this limitation, [7] proposed a hypergraph structure to capture the query operation sharing among very large sets of a priori known queries. Encouraging results have been obtained showing the benefit of hypergraphs to coupling these two problems [7] [22].

In this study, we capitalize on the usage of hypergraphs to deal jointly with $PMQO$ and the VSP, by considering at the same time the 4-properties. The dynamic aspect imposes us to incrementally construct our hypergraph. To do so, a set of primitives is defined on hypergraph allowing its construction in an incremental way and capturing the query interaction. During the construction of our hypergraph, a set of materialized views is selected. These views do not have an infinite lifetime, but they can be replaced by others in order to reduce the query processing cost. An example of the connectivity of our proposal with Oracle DBMS is given.

The remainder of this paper is organized as follows: Section 2 proposes a concise state-of-art related to our two studied problems. Section 3 introduces fundamental notions and definitions related to hypergraphs. Section 4 details our contribution and highlights the main steps of our dynamic process. Section 5 validates our proposals. Section 6 concludes the paper.

2 RELATED WORK

In this section, we overview the major studies dealing with $PMQO$ and VSP in isolated and joint ways and their role in solving other important problems.

$PMQO$ is one of the fundamental topics in databases [23]. In the Encyclopedia of Database Systems [19], a specific chapter on this topic has been reserved. To position the $PMQO$ according to the VSP, its generic static formalization is needed. For a given a set of queries $\{Q_1, \ldots, Q_n\}$ to be optimized, where each query Q_i has a set of possible individual plans, the $PMQO$ consists in finding a global query plan by merging individual plans such that the query processing cost of executing all queries is minimized. [25] showed that $PMQO$ problem is NP-hard and gave its state-space search formulation. Several variations of Sellis's algorithms have been proposed [25] either to handle larger $PMQO$ problems [26] or for extending top-down cost-based query optimizers to support multi-query optimization [23].

In the physical design phase that has been amplified by the explosion of the DW technology, materialized views are a serious solution to optimize complex OLAP queries and well connected to query operation sharing. Surprisingly, the VSP has a quite similar formalization as the $PMQO$: given a set of queries $\{Q_1, \ldots, Q_n\}$ and a set of constraints C (e.g., storage cost, maintenance cost, processing cost), the VSP consists in selecting a set of materialized views that satisfy the fixed non-functional requirements such as query performance, energy consumption, etc. and respects C. The selection of these views is guided by the number of sharing operations among queries. The VSP is known as an NP-hard problem [14]. Several studies have been proposed to deal with this problem. We recommend the readers to refer to [22] which give a nice classification of the existing studies. Most of the existing studies consider a reasonable set of static queries. In order to track the changes in query workload, the DynaMat is one of the fundamental systems proposed in [18]. It uses a pool that materializes fragments representing only aggregate query results from incoming queries and stores them in a pool and exploits them for the future; by taking time and space constraints. This work does not refer to any paper related to $PMQO$.

By deeply examining the VSP studies, we figure out that they do not interact with $PMQO$ [28]. Some do not even cite the pioneering article of $PMQO$ [24], except the work of [30] has highlighted this dependency. The authors proposed a bottom-up scenario to construct a global plan of queries. Initially, the authors select the join local plans of each query (logical plans that have only join operations). These plans are merged in a single unified query plan ($MVPP$), represented by an acyclic graph. Two algorithms are proposed for selecting the best $MVPP$ which has the minimum cost. The first algorithm called A feasible solution generates all possible $MVPP$ and the plan with the minimum cost will be chosen. This algorithm is costly in terms of computation. To simplify the previous algorithm, a second algorithm is proposed based on 0-1 integer programming. The view selection algorithm is performed in two steps: (1) generation of materialized views candidates that have positive benefits between query processing and view maintenance. This benefit corresponds to the sum of query processing using the view minus the maintenance cost of this view, (2) only candidate nodes with positive benefits are selected to be materialized.

The main shared point by $PMQO$ and VSP problems is that they use graph data structures either to identify the sub-common expressions among queries or to prune the research space of these problems. VSP studies used three main graph structures: AND/OR view-graph [14] , $MVPP$ [30] and data cube lattice [15]. In $PMQO$, connection graphs or query graphs have been used in [11] to represent queries involving Select, Project, and Join operators. In a query graph, one node indicates the result relation, and any other node indicates an operand relation. An edge in the connection graph represents a join in the case where both of connection nodes are operand relation, else it represents a projection. The fact that an edge in a connection graph may be associated with more than two nodes making it a hypergraph that has been exploited to deal with the joint problem in [6].

The $PMQO$ has been exploited to manage *concurrent* large-scale data mining queries submitted to a Knowledge Discovery Management System to select Frequent Item Sets (FIS) [29]. In this work, the FP-Growth algorithm that uses the FP-tree structure has been proposed. This work is quite similar to ours since it considers two properties of the data mining queries (large-scale and interacted)

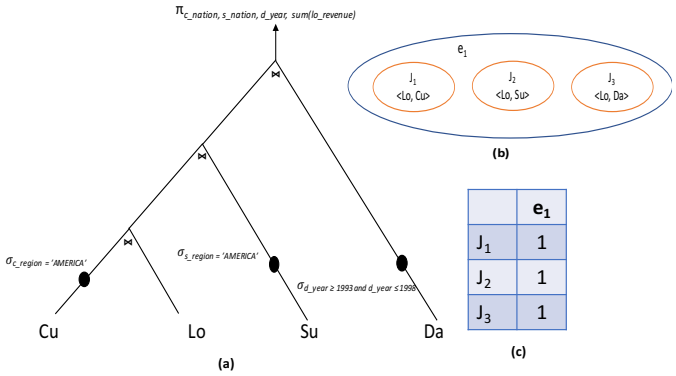


Figure 1: Query Tree of Q (a), its Hypergraph (b) and Incidence Matrix (c).

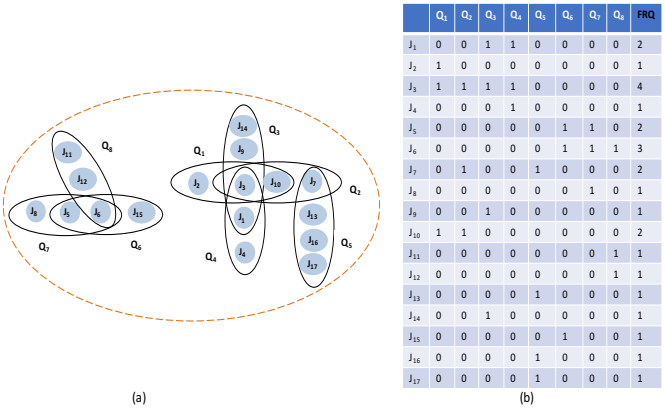


Figure 2: Global Hypergraph for 8 Queries.

and uses a graph-like data structure. But is ignores the two other properties. Another difference concerns the no-similarity between mining and OLAP queries.

3 HYPERGRAPHS AS A SERVICE FOR MANAGING 4-PROPERTIES

In this section, fundamental notions related to hypergraphs are first presented and then the ability to manage workload with 4-properties.

Definition 3.1. A hypergraph [8] is a pair $H = (V, E)$, where $V = \{v_1, \ldots, v_n\}$ is a set of vertices (or nodes) and $E = \{e_1, \ldots, e_m\}$, where $e_j \subseteq V$ is a set of hyper-edges [8]. Clearly when $|e_i|= 2$ $(\forall i = 1...m)$, the hypergraph is a standard graph.

Definition 3.2. The degree of a vertex $v_i \in$V, denoted by $d(v_i)$ is defined as the number of distinct hyper-edges in E that connect v_i.

Example 3.1. To show how an OLAP query can be represented by a hypergraph, let us consider the following query Q defined on the star schema benchmark (SSB)[4] that contains a fact table *Lineorder* and four dimension tables *Customer*, *Supplier*, *Part*, and *Dates*.

```
Q: Select cu_nation, su_nation, da_year, sum(lo_revenue) as revenue
from Customer Cu, Lineorder Lo, Supplier Su, Dates Da
where lo_custkey = c_custkey (J1)
and lo_suppkey = s_suppkey (J2) and lo_orderdate = da_datekey (J3)
and c_region = 'AMERICA' and s_region = 'AMERICA'
and d_year >= 1993 and d_year <= 1998
group by c_nation, s_nation, d_year
order by d_yearasc, revenue desc;
```

The query graph of Q and its hypergraph are given in Fig. 1. As we see, the hypergraph contains three vertices representing the three joins (J_1, J_2, J_3) and one hyperedge e_1 corresponding to our query (Q). *For simplicity, we ignore selection operations and we focus on joins because they are most costly and automatically candidate for materialization*. For visibility, the join nodes (vertices) are identified by a join predicate and two selection predicates, one selection on the fact table and the other on the dimension table. As any graph, our hypergraph can easily be represented by an incidence matrix IM, whose rows and columns represent respectively vertices and hyperedges. The (i,j)th entry in the matrix, denoted by IM_{ij} is equal to 1 if vertex v_i is contained in the hyperedge e_j, and is 0 otherwise. The column representing the query is called *query vector* characterizing its joins.

[4] http://www.cs.umb.edu/~poneil/StarSchemaB.pdf

3.1 Hypergraph for managing query interaction

The global hypergraph (GH) dedicated to representing a workload of queries follows the same above principles, by capturing the query operation sharing. For a workload of queries W, a hypergraph GH is a set of vertices GV and a set of hyperedges GE, where GV represents a set of join nodes, such that for each vertex $v_i \in GV$, corresponds a join node n_i. The same way, GE represents the workload W, such that for each hyperedge $e_j \in GE$ corresponds to the query Q_j. A hyperedge e_j connecting a set of vertices corresponds to join nodes that participate in the execution of the query Q_j.

Example 3.2. To illustrate this construction, let us consider 8 queries generated randomly using the SSB query generator. Fig. 2 shows the global hypergraph that captures the interaction between queries.

This matrix is quite similar to transactional matrix used for detecting frequent item sets (FIS) [1]. We add a new column representing the usage frequency of each join operation J_i, denoted by FRQ_i. It is computed as follows: $\sum_{j=1}^{n} IM_{ij}$.

Our global hypergraph contains two disjoint components: $Comp_1$: (Q_6, Q_7, Q_8) and $Comp_2$: $(Q_1, Q_2, Q_3, Q_4, Q_5)$. We observe that queries $Q_6, Q7$, and Q_8 of $Comp_1$ share the same join (J_6) that is a candidate for materialization. The incidence matrix gives several hints in materializing views as FIS did [29] for the same purpose.

3.2 Hypergraph for managing query volume

The database technology has a great experience in using advanced data structures such as graphs, hypergraphs, trees for either to speed up the data access or as a support for algorithms dealing with large scale search space. Hypergraphs are more expressive than traditional graphs in modeling sets of objects since they contribute in capturing any relationship between a group of objects, whereas graphs can only capture binary relationships [10]. This specificity gives more flexibility in accurately formulating several important problems in combinatorial scientific computing [31]. Another particularity that

hypergraphs offer is their ability in dividing large search space of a complex problem into several sub search spaces through partitioning. Hypergraphs show their benefit in many application domains such as very large scale integration (VLSI), where they represent very large circuits.

4 DYNAMIC CONSTRUCTION OF HYPERGRAPH AND SELECTION OF VIEWS

Example3.2 showed the construction of hypergraph and its usage in selecting materialized view candidates per component in a static way. These two processes have to be revisited to consider the context of our 4-properties. This necessitates an incremental construction of global hypergraph, dynamic selection of views, and good management of the pool of views associate with each hypergraph component as shown in Fig. 3. Different modules of our proposal are discussed in the next sections.

Queries are coming dynamically in text format and then parsed. After parsing it, its graph is constructed (see Example 3.1). To ensure the incremental construction of our global hypergraph GH, we define primitives which are: *add an edge*, *add a node*, *remove an edge*, and *remove a node*. They are specified as follows:

- $Add\_hyperedge(GH_t, Q_t) \rightarrow GH_{(t+1)}$: adds the hyperedge Q_t (query) of the hypergraph GH_t at instant t and as an output the hypergraph $GH_{(t+1)}$ (at instant $(t+1)$) is obtained;
- $Add\_node(GH_t, J_t) \rightarrow GH_{(t+1)}$: adds the join J_t (join) to the hypergraph GH_t and as an output, the hypergraph $GH_{(t+1)}$ is obtained;
- $Remove\_hyperedge(GH_t, Q_t) \rightarrow GH_{(t+1)}$: removes the edge Q_t from the hypergraph GH_t and as an output, the hypergraph $GH_{(t+1)}$ is obtained;
- $Remove\_node(GH_t, J_t) \rightarrow GH_{(t+1)}$: removes the node J_t from the hypergraph GH_t, and as an output, the hypergraph $GH_{(t+1)}$ is obtained.

This incremental construction impacts the static structures above presented. At each instant t, incidence matrix of the GH is defined as in the static way. Component representation and size will also change dynamically. Here the process of managing the arrival of queries:

(1) When the first query Q_f arrives, it is parsed and its join(s) are identified (denoted by $Join(Q_f)$). This query contributes in constructing the first component of the global hypergraph using the primitive $Add\_hyperedge(GH_f, Q_f)$.

(2) When the second query Q_s is received, we test if it shares join nodes of the query Q_f. (1) If $Join(Q_s) \bigcap Join(Q_f) \neq \phi$ then Q_s is then placed in the first component using the $add\_hyperedge$ primitive. After that, we compute the benefit of each join of Q_s. We define the benefit of a node J_i by benefit(J_i) using the same formulation of [7], which take into account the benefit of the join node and the maintenance cost caused by its materialization:

$$benefit(n_i) = (nbr\_use - 1) \times processing\_cost(J_i) - const\_cost(J_i) \quad (1)$$

where nbr_use, $processing\_cost(J_i)$, and $const\_cost(J_i)$ represent respectively the number of queries that use the join node J_i, the processing cost of J_i, and the construction cost of J_i.
Therefore, the joins with a positive benefit will be materialized. (2) In the case when Q_s does not share any join with Q_f ($Join(Q_f) \bigcap Join(Q_s) = \phi$), a new **component** is constructed and associated with the query Q_s.

(3) When the third query Q_{rd} arrives, its placement process is based on a criterion defined between the query Q_{rd} and the existing components (hypergraphs) of queries. In order to reuse the existing materialized views, the query Q_{rd} has to be placed in the component that shares with it the maximum of join operations already materialized. In the case when Q_{rd} does not share any materialized join with the existing components, we place Q_{rd} in the component that shares with it the maximum of join operations. If Q_{rd} does not share any join, a new hypergraph is constructed and associated with the query Q_{rd}.

We would like to mention that every time we add a new query to a hypergraph, we re-compute the benefit of materializing joins that appear in the query. After that, we test if there are new join nodes with a positive benefit computed using Equation 1. Algorithm 2 is called that updates the pool of materialized views selected for this component by adding joins that have a positive benefit. This process will be repeated at each arrival of a new query to a hypergraph component until the saturation of storage space. If the storage disk space is full, we replace the least beneficial materialized views of this component by joins that have a positive benefit in the current query.

Example 4.1. Let us consider the nine queries described in *Appendix 1* . We assume that their arrival follows the following order: $Q_1 \rightarrow Q_2 \rightarrow Q_3 \rightarrow Q_4 \rightarrow Q_5 \rightarrow Q_6 \rightarrow Q_7 \rightarrow Q_8 \rightarrow Q_9$. Fig. 4 shows the incremental construction of the hypergraph components and the dynamic selection of materialized views for these 9 queries. Four components were constructed incrementally. Three joins with positive benefits that are selected as materialized views as followed: J_1 and J_3 belonging to the component 1 and J_8 belonging to the component 3.

5 EXPERIMENTAL STUDY

The main goal of this section is twofold: (i) showing the connection of our proposal to a commercial DBMS and (ii) validation and comparison of our proposal with existing state-of-art studies.

5.1 Feasibility study: HYRAQ Connection to Oracle DBMS

We developed a tool, called HYRAQ, supporting our proposal. It is developed using Java and integrates all modules of our proposal

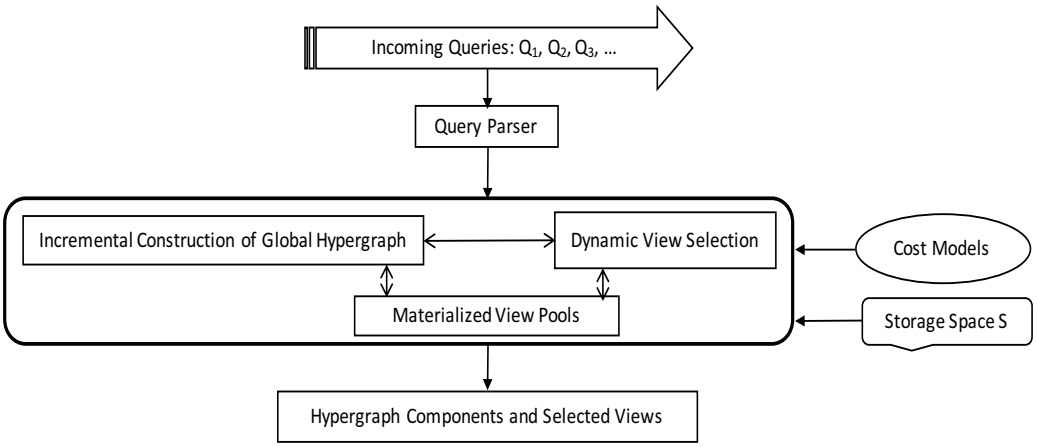


Figure 3: The global architecture of our Proposal

Algorithm 1 Dynamic Materialization and Hypergraph Incremental Construction

Inputs: an incoming query Q^t_{icom} at instant t; Storage space S;
Outputs: a list of components ($Comp$) of GH; a set of pools of views per component;
begin
loop
$t := 1$;
Parse_Query (Q^t_{icom});
if $|Comp^t| = 0$ **then**
  $comp^t := Create\_new\_component()$;
  $add\_hyperedge(comp^t, Q^t_{icom})$;
  $IM^t := Create\_incidence\_matrix(GH^t)$;
else
  for each $Comp^t_i \in GH^t$ **do**
    $Nbr\_shared\_views^t := |Joins(Q^t_{icom}) \bigcap Pool^{comp^t_i}|$;
    $List^t_1.add(Nbr\_shared\_views^t)$;
    $Nbr\_of\_shared\_Joins^t := |Joins(Q^t_{icom}) \bigcap comp^t_i.GV^t|$; $\{GV^t$: vertices of $comp^t_i\}$
    $List^t_2.add(Nbr\_of\_shared\_Joins^t)$;
  end for
  $Max^t_1 := Max(List^t_1)$;
  if $Max^t_1 = 0$ **then**
    $Max^t_2 := Max(List^t_2)$;
    if $Max^t_2 = 0$ **then**
      $Create\_new\_component(comp^t_i)$;
      $add\_hyperedge(comp^t_i, Q^t_{icom})$;
      $Comp.add(comp^t_i)$;
      $IM^t := Update\_incidence\_matrix$;
    else
      $position^t := Return\_position(Max^t_2, List^t_2)$;
      $add\_hyperedge(Comp.get(position^t), Q^t_{icom})$;
      $IM^t := Update\_incidence\_matrix$;
      $Dynamic\_Views\_Selection(Q^t_{icom}, Comp.get(position^t), Pool^{Comp.get(Position^t)}, S)$;
    end if
  else
    $Position^t := Return\_position(Max^t_1, List^t_1)$;
    $add\_hyperedge(Comp.get(position^t), Q^t_{icom})$;
    $Rewrite(Q^t_{icom}, Pool^{Comp.get(Position^t)})$; {Query writing using $comp^t_i$'s views.}
    $IM^t := Update\_Incidence\_Matrix$;
    $Dynamic\_Views\_Selection(Q^t_{icom}, Comp.get(Position^t), Pool^{Comp.get(Position^t)}, S)$;
  end if
end if
$t := t + 1$;
END LOOP

(Fig. 5). It monitors permanently the coming queries and it is connected to Oracle DBMS. At the arrival of a new query, HYRAQ sends it to Oracle with its appropriate views. HYRAQ plays the role of any DBMS advisor with a strong particularity of managing 4-properties workloads.

5.2 Efficiency Study

We conducted several experiments to evaluate the efficiency of our proposal against major state-art studies. Before describing and

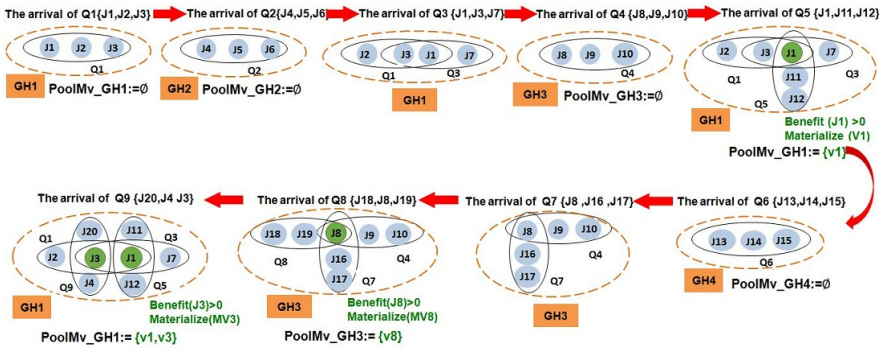


Figure 4: Dynamic construction of global hyper graph and selection of views.

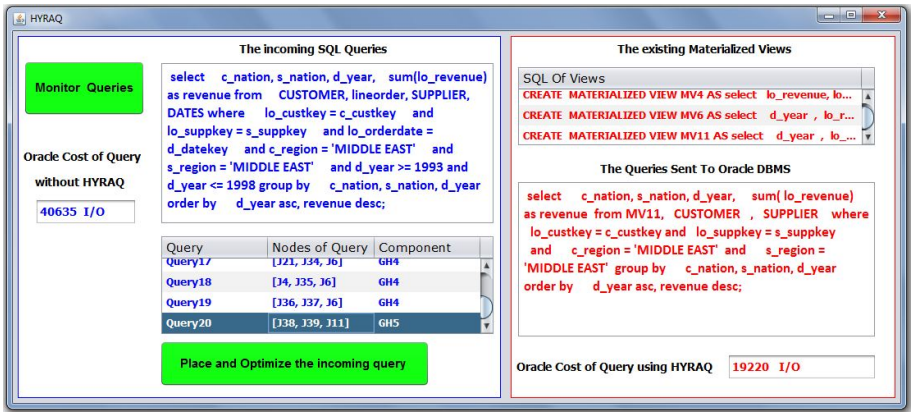


Figure 5: An Example of Functioning of HYRAQ.

Algorithm 2 Dynamic_Views_Selection

Inputs an incoming query: Q_{icom}; a component (sub-hypergraph): $compi$; the pool of Materialized Views of the Component compi: $Pool$; Storage Space: S
Output the updated pool Materialized Views of the component compi: $Pool$
begin
$joins \leftarrow Join(Q_{icom})$
Calculate_Benefit (Joins)
$L \leftarrow Return\_Joins\_with\_Positive\_Benefit(Joins)$
DescendingOrder(L) {according to the benefit}
for each node in L **do**
  if $node \notin Pool^{compi}$ $And\ size(Pool^{compi}) + size(node) < S$ **then**
    Materialize (node)
    $Pool^{compi}.add(node)$
    $size(Pool^{compi}) \leftarrow size(Pool^{compi}) + size(node)$
  else
    if $node \notin Pool^{compi}$ $And\ size(Pool^{compi}) + size(node) > S$ **then**
      Ascending Order($Pool^{compi}$) {according to the benefit}
      $index \leftarrow 0$
      repeat
        if $benefit(node) > benefit(Pool[index])$ **then**
          Drop (Pool [index])
          $size(Pool^{compi}) \leftarrow size(Pool^{compi}) - size(node)$
          $index \leftarrow index + 1$
        end if
      until $size(Pool^{compi}) + size(node) < S$ OR $benefit(node) < benefit(Pool[index])$
      if $size(Pool^{compi}) + size(node) < S$ **then**
        Materialize (node)
        $Pool^{compi}.add(node)$
        $size(Pool^{compi}) \leftarrow size(Pool^{compi}) + size(node)$
      end if
    end if
  end if
end for

commenting on them, we present the environment and data set that we used to perform these experiments. All experiments were performed on a server with Intel Xeon E5620 2.40 GHz processor, 32 GB of main memory, and 1 TB of the hard disk. We generate a DW with 30 GB deployed in Oracle 12c DBMS and queries using SSB generator modules.

Our proposal is compared with two algorithms: **(1)** the algorithm proposed in [7] (that we name it *StaticHypergraph*) that uses hypergraphs to capture query volume and interaction, but deals only with static workloads. Our choice is based on the important finding of *StaticHypergraph* that outperforms the approach proposed by [20] that dynamically materializes query views by scheduling queries in order to augment the benefit of selected materialized views. Query scheduling avoids massively view dropping. In other words, when a view is materialized, it should optimize the maximum of queries before its dropping. (2) The work proposed in [29] (that we name it *FIS Algorithms*) that brings the PMQO concepts to the field of data mining. This work is quite similar to our work since it considers two properties (large-scale and interacted) of data mining queries and uses a graph-like data structure. This work has used the FP-Growth algorithm which aims to extract frequent itemset (FIS) based on a compact data structure called FP-Tree. We aim by this choice to compare the efficiency of hypergraphs and its alternative in the field of FIS mining.

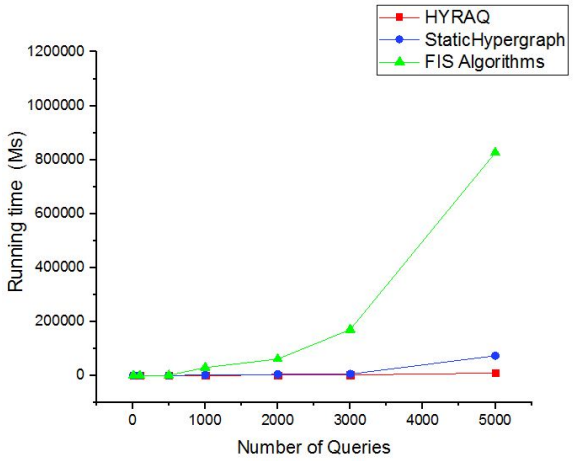


Figure 6: Scalability Comparison

To adapt *FIS Algorithms* to the context of analytical queries, we consider the following steps for a static set of queries:

(1) **Parsing queries:** by using our parser module.
(2) **Scheduling queries:** due to the large search space of PMQO problem, several heuristics have been proposed to prune the search space in the filed of FIS mining. The used heuristic algorithm in our experiments is CCGreedy algorithm [4]. Our choice is motivated by the fact that the CCGreedy algorithm is proved to be sound in several experiments and it offers a significant improvement in accuracy over the existing heuristics and provided acceptable running times. Also, it facilitates the construction of the unified query plan of each component in (step4), this is due to the use of hypergraphs to partition the set of queries into several components.
(3) **Selection of the candidates views for each component:** We use the FP-Growth algorithm (see [29] for an overview) to identify the most shared intermediate results For each component of queries. To do so, we assimilate the intermediate results (join operations) to frequent itemsets. Therefore, the selected joins are those with a frequency threshold greater than the fixed minimum support threshold (Minsup) (the used Minsup in our experiments is 3). The selected nodes are considered as candidates for materialization.
(4) **Generating the unified query plan for each component**. We mention that we have used for this step the same method of [7]. This step allows ordering nodes and materializing the candidate views selected in **Step 3**.

5.2.1 Scalability of the Algorithms. This experiment aims to study the scalability of the different algorithms according to the number of received queries. To do so, for each test, we change the size of the query workload and we compute the execution time of the three programs. The obtained results are depicted in Fig.6. We observe that the two algorithms, which are based on hypergraphs outperform FIS algorithms. Moreover, the results reveal that HYRAQ is faster than the algorithm based on StaticHypergraph.

5.2.2 Number of materialized views and their benefit. This experiment aims to study the number of materialized views generated by the studied algorithms and their benefit on queries. To do so, two experiments were conducted by considering both static and dynamic workloads with 100 and 1 000 queries. The obtained results are summarized in Fig. 7. They show that *StaticHypergraph* outperforms *HYRAQ*. This is because *StaticHypergraph* knows in advance the workload, which is an unrealistic hypothesis. They show also that FIS algorithms and *HYRAQ* have optimized almost the same number of queries. Although, FIS algorithms know in advance the workload of queries contrary to *HYRAQ* that selects fewer views than the other algorithms.

5.2.3 Impact of selected views on query processing and construction costs. This experiment has the goal to study the contributions of the selected materialized views by the three algorithms on overall query processing and their construction costs using the same above scenario. To do so, we evaluate theoretically the different costs using the mathematical cost model developed in [5]. Our choice of this cost model is based on its high quality proven in different studies [7, 17, 21]. The obtained results are reported in Fig. 8. The selected views by *StaticHypergraph* are more beneficial than those generated by *HYRAQ*. It showed also that FIS algorithms outperform slightly *HYRAQ* in optimizing the overall processing cost of queries. However, our algorithm is better than others in terms of the consumed construction cost. This is due to our materialization strategy that materializes just the most interesting views, which can be used by future queries.

5.2.4 Impact of selected views on the storage space. In the third experiment, we study the total storage resource fixed to 500 GB consumed by the three algorithms. Fig. 9 shows that *HYRAQ* consumes less storage space than FIS and *StaticHypergraph* algorithms since it materializes less and beneficial views that reduces their storage space.

5.2.5 Materializing or not strategies on optimizing queries. In the last experiment, we first evaluate theoretically (using the mathematical cost model) the benefit of materializing views on the reduction of the processing cost of queries and compare it with the scenario where none view is materialized by considering static and dynamic workloads. Static workloads allow us to evaluate *StaticHypergraph* and FIS algorithms. We define the reduction rate as:

$$1 - \frac{query\ cost\ with\ views}{query\ cost\ without\ views}.$$

Fig.10a shows the obtained results using the mathematical cost model.

To confirm our theoretical results, we have implemented in Oracle 12c DBMS. Algorithm 3 describes the used steps in connect *HYRAQ* to Oracle for calculating the global Oracle cost of the incoming queries. The obtained results in Fig. 10b show that *StaticHypergraph* outperforms the other algorithms. Our approach becomes more interesting when the number of queries increases (cost reduction rate of 58% after receiving 1000 queries). This is due to the expansion of our pool by the most beneficial materialized views. The obtained results follows those obtained with the mathematical cost models.

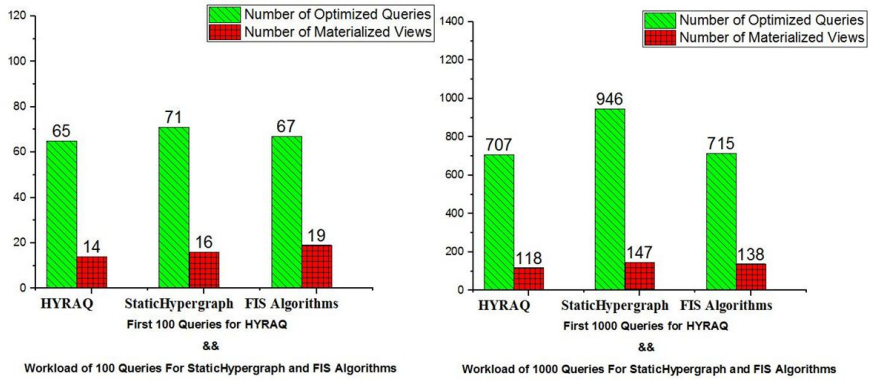


Figure 7: A comparison between *HYRAQ*, StaticHypergraph and FIS algorithms

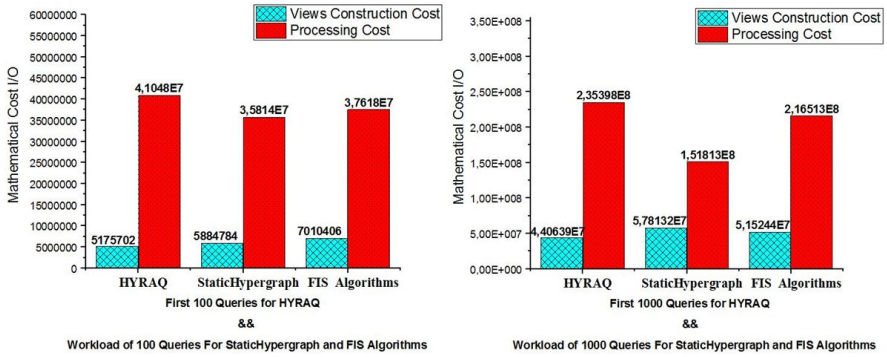


Figure 8: Comparison among the three approaches in terms of processing/maintenance costs.

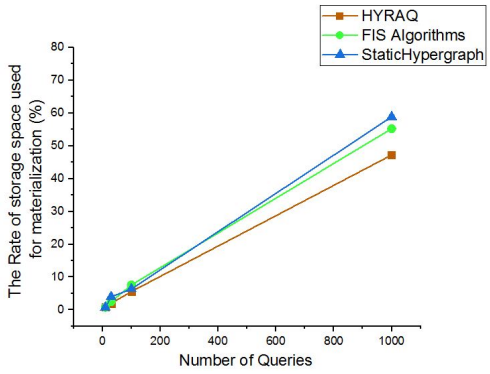


Figure 9: The consumed storage space by the three algorithms

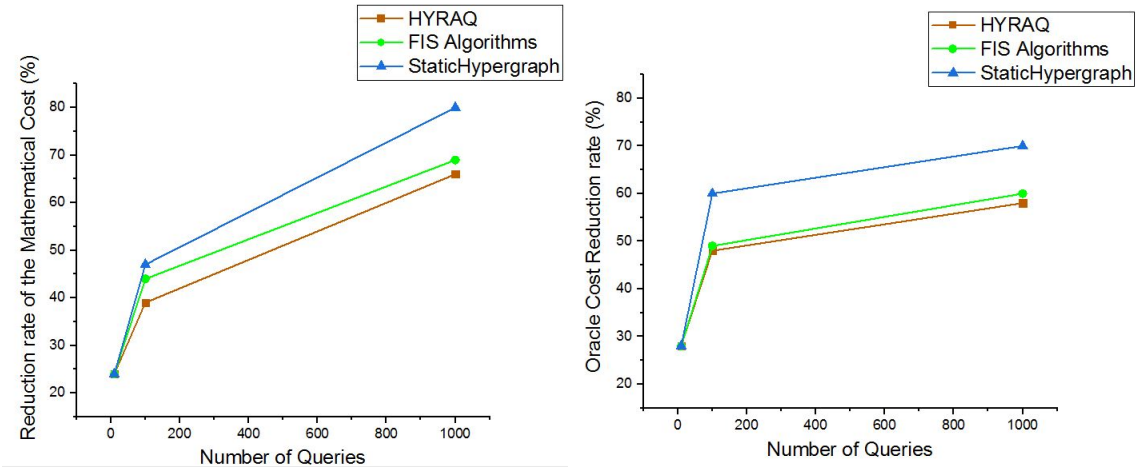
Algorithm 3 CALCULATE_ORACLE_COST

Inputs an incoming query Q^t_{icom} at instant t
Outputs $Global\_oracl\_cost\_using\_HYRAQ, Global\_oracl\_cost\_without\_HYRAQ$

begin
loop
$t := 1$;
$OracleCost := estimatetheOracleCostof(Q^t_{icom})$ {using EXPLAIN PLAN}
$Global\_cost\_without\_HYRAQ := Global\_cost\_without\_HYRAQ + OracleCost$
if materialized view(s) exist for a Q^t_{icom} **then**
  Rewrite(Q^t_{icom}) {Manual Rewritting of Q^t_{icom} using the existing Views}
  $OracleCost := estimatetheOracleCostof(Q^t_{icom})$ {using EXPLAIN PLAN}
  $Global\_cost\_using\_HYRAQ := Global\_cost\_using\_HYRAQ + OracleCost$
else
  $OracleCost := estimatetheOracleCostof(Q^t_{icom})$ {using EXPLAIN PLAN}
  $Global\_cost\_using\_HYRAQ := Global\_cost\_using\_HYRAQ + OracleCost$
end if
$t := t + 1$;
END LOOP

6 CONCLUSIONS

In this paper, we revisit the two well known studied and *interconnected* problems in the database world which are: *multi-query optimization* and *physical design* under a new angle, by considering analytical queries running on modern applications. It should be noticed that in the literature, mastering the interaction between these two problems contributes to solving several problems in physical design such as selecting materialized views. Nowadays, analytical queries are (1) voluminous, (2) ad-hoc, (3) dynamic, and (4) share similar operations. The traditional solutions cannot be reproduced directly to optimize these queries either at the logical level or at the physical level (e.g., by selecting materialized views). We claim that optimizing these queries using physical techniques requires scalable data structures such as hypergraphs that already showed their contributions for static queries. To deal with our queries with their 4-properties, we proposed the use of *dynamic hypergraphs* to capture easily the interaction among arrival queries and to ease the



(a) mathematical cost reduction rate

(b) Oracle cost reduction rate

Figure 10: Theoretical and real validation of the studied algorithms

process of dynamically selecting materialized views. Our solution consists in materializing the most shared join operations dynamically identified by our hypergraph. Another particularity of our proposal is that the selected views do not have an infinite lifespan since they can be deleted and replaced by others based on their benefit in optimizing the current queries. This benefit is calculated using mathematical cost models including query processing, view storage, and maintenance costs. Our finding is compared against two types of existing methods: those using only static hypergraphs and those using data mining techniques. Our proposal is supported by a tool called HYRAQ that can be connected to any DBMS. In our experiment, HYRAQ is linked to Oracle DBMS, where its theoretical results are directly deployed in that DBMS. Our experiments and tool show the efficiency and effectiveness of our study.

Our work opens several issues: **(i)** conducting other experiments to compare the efficiency of HYRAQ against commercial advisors, **(ii)** integrating query scheduling in HYRAQ to manage incoming queries and increase their interaction, **(iii)** considering other optimization structures such as indexes, horizontal data partitioning and **(iv)** reproduce our proposal to SPARQL queries over Linked Open Data.

REFERENCES

[1] R.Agrawal and R.Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*,pages 487–499.ACM, 1994.

[2] L.Bellatreche and A.Kerkad. Query interaction based approach for horizontal data partitioning. *IJDWM.*, 11(2): 44-61 ,2015

[3] D.Benvenuto, M.Giovanetti, L.Vassallo, S.Angeletti and M.Ciccozzi. Application of the arima model on the covid-2019 epidemic dataset. *Data in Brief* ,2020.

[4] P.Boinski, M.Wojciechowski and M.Zakrzewicz. A greedy approach to concurrent processing of frequent itemset queries. In *DaWaK*, pages 292-301.Springer, 2006.

[5] A.Boukorca. *Hypergraphs in the Service of Very Large Scale Query Optimization. Application*. Phd thesis, Ecole nationale supérieure de mécanique et d'aérotechnique, Chasseneuil-du-Poitou, Poitiers, 2016.

[6] A.Boukorca, L.Bellatreche and A.Cuzzocrea. SLEMAS: an approach for selecting materialized views under query scheduling constraints. In *COMAD*, pages 66-73. Computer Society of India, 2014.

[7] A.Boukorca, L.Bellatreche, S.B.Senouci and Z.Faget. Coupling materialized view selection to multi query optimization: Hyper graph approach. *IJDWM.*, 11(2): 62-84 ,2015.

[8] A.Bretto. *Hypergraph Theory: An Introduction*. Springer, 2013.

[9] J.Camacho-Rodríguez , et al. Apache hive: From mapreduce to enterprise-grade big data warehousing. In *SIGMOD*, pages 1773-1786. ACM, 2019.

[10] V.Ü.Çatalyürek and C.Aykanat. Decomposing irregularly sparse matrices for parallel matrix-vector multiplication. In *Third International Workshop on Parallel Algorithms for Irregularly Structured Problems (IRREGULAR)*, pages 75-86. Springer, 1996 .

[11] U.S.Chakravarthy and J.Minker. Multiple query processing in deductive databases using query graphs. In *VLDB*, pages 384-391. ACM, 1986.

[12] J.Chen and H.Wang. Guest editorial: Big data infrastructure I. *IEEE Trans. Big Data.*,4(2): 148-149, 2018.

[13] E.Dong, H.Du, and L.Gardner. An interactive web-based dashboard to track covid-19 in real time. *The Lancet infectious diseases.* ,2020.

[14] H.Gupta and I.S.Mumick. Selection of views to materialize under a maintenance cost constraint. In *ICDT*,pages 453-470. Springer, 1999.

[15] V.Harinarayan, A.Rajaraman and J.D.Ullman. Implementing data cubes efficiently. In *ACM SIGMOD*,pages 205-216. ACM, 1996.

[16] S.Jain, D.Moritz, D.Halperin, B.Howe and E.Lazowska. Sqlshare: Results from a multi-year sql-as-a-service experiment. In *ACM SIGMOD* ,pages 281-293. ACM, 2016.

[17] A.Kerkad, L.Bellatreche and D.Geniet. Queen-bee: Query interaction-aware for buffer allocation and scheduling problem. In *DaWaK*,pages 156-167. Springer, 2012.

[18] Y.Kotidis and N.Roussopoulos. Dynamat: A dynamic view management system for data warehouses. In *ACM SIGMOD*,pages 371-382. ACM, 1999.

[19] L.Liu and M.T. Özsu., Eds. *Encyclopedia of Database Systems, 2nd Edition*. Springer, 2018.

[20] T.Phan and W.Li. Dynamic materialization of query views for data warehouse workloads. In *ICDE*,pages 436-445. IEEE, 2008.

[21] R.Bouchakri and L.Bellatreche. On simplifying integrated physical database design. In *ADBIS*,pages 333-346. Springer, 2011.

[22] A.Roukh, L.Bellatreche, S.Bouarar, and A.Boukorca.. Eco-physic: Eco-physical design initiative for very large databases. *Information Systems, 68:* 44-63, 2017.

[23] P.Roy and S.Sudarshan. Multi-query optimization. In *in [19]*, 2018.

[24] T.Sellis. Multiple-query optimization. *ACM TODS* , 13(1): 23-52, 1988.

[25] T.Sellis and S.Ghosh. On the multiple query optimization problem. *IEEE Transactions on Knowledge and Data Engineering* , 2(2): 262-266 ,1990.

[26] K.Shim, T.K.Sellis and D.S.Nau. Improvements on a heuristic algorithm for multiple-query optimization. *Data Knowl. Eng.* 12(2): 197-222 , 1994

[27] B.Skiera, L.Jrgensmeier, K.Stowe and I.Gurevych. How to best predict the daily number of new infections of covid-19. *arXiv e-prints* ,2020.

[28] M.K.Sohrabi and H.Azgomi. Evolutionary game theory approach to materialized view selection in data warehouses. *Knowl. Based Syst.* 63: 558-571 , 2019

[29] M.Wojciechowski, K.Galecki and K.Gawronek. Three strategies for concurrent processing of frequent itemset queries using fp-growth. In *KDID*, pages 240-258. Springer, 2006.

[30] J.Yang, K.Karlapalem and Q.Li. Algorithms for materialized view design in data warehousing environment. In *VLDB*, pages 136-145, 1997

[31] A.Yzelman and R.H.Bisseling. Cache-oblivious sparse matrix-vector multiplication by using sparse matrix partitioning methods. *SIAM Journal on Scientific*

Computing. 31(4): 3128-3154 ,2009.

Appendix 1:

```
Q1: Select sum(lo_revenue), d_year,
p_brand
from lineorder, dates,part,supplier
where lo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_category = 'MFGR#34'
and s_region = 'EUROPE'
group by d_year, p_brand
order by d_year, p_brand;

Q2: Select c_city, s_city, d_year,
sum(lo_revenue) as revenue
from customer,lineorder,supplier,dates
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and (c_city='ALGERIA 3'
or c_city='VIETNAM 8')
and (s_city='ALGERIA 3'
or s_city='VIETNAM 8')
and d_yearmonth = 'Mar1993'
group by c_city, s_city, d_year
order by d_year asc, revenue desc;

Q3: Select sum(lo_revenue),d_year,
p_brand
from lineorder,dates,part,supplier
wherelo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_brand between 'MFGR#3334'
and 'MFGR#3410'
and s_region = 'EUROPE'
group by d_year, p_brand
order by d_year, p_brand;

Q4: Select c_nation,s_nation,
d_year,
sum(lo_revenue) as revenue
from customer,lineorder, supplier,dates
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and c_region = 'MIDDLE EAST'
and s_region = 'MIDDLE EAST'
and d_year >= 1993 and d_year <= 1998
group by c_nation, s_nation, d_year
order by d_year asc, revenue desc;

Q5: Select sum(lo_revenue),d_year,
p_brand
from lineorder, dates, part,supplier
where lo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_category = 'MFGR#13'
and s_region = 'ASIA'
group by d_year, p_brand
order by d_year, p_brand;

Q6: Select c_city,s_city,d_year,
sum(lo_revenue) as revenue
from customer,lineorder,supplier,dates
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and (c_city='IRAQ 1'
or c_city='JAPAN 5')
and (s_city='IRAQ 1'
or s_city='JAPAN 5')
and d_yearmonth = 'Jun1993'
group by c_city, s_city, d_year
order by d_year asc, revenue desc;

Q7: Select c_city, s_city, d_year,
sum(lo_revenue) as revenue
from customer,lineorder,supplier,dates
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and (c_city='MOROCCO 1'
or c_city='SAUDI ARA1')
and (s_city='MOROCCO 1'
or s_city='SAUDI ARA1')
and d_year >= 1993 and d_year <= 1998
group by c_city, s_city, d_year
order by d_year asc, revenue desc;


Q8: Select c_city, s_city, d_year,
sum(lo_revenue) as revenue
from customer,lineorder,supplier,dates
where lo_custkey = c_custkey
and lo_suppkey = s_suppkey
and lo_orderdate = d_datekey
and c_nation = 'GERMANY'
and s_nation = 'GERMANY'
and d_year >= 1993 and d_year <= 1998
group by c_city, s_city, d_year
order by d_year asc, revenue desc;

Q9: Select sum(lo_revenue),d_year,
p_brand
from lineorder, dates, part,supplier
where lo_orderdate = d_datekey
and lo_partkey = p_partkey
and lo_suppkey = s_suppkey
and p_brand between 'MFGR#5325'
and 'MFGR#5332'
and d_year = 1994 or d_year = 1996
and s_region = 'EUROPE'
group by d_year, p_brand
order by d_year, p_brand;
```

Benchmarking a Distributed Database Design that Supports Patient Cohort Identification

Jero Mario Schäfer
Institute of Computer Science
Department of Mathematics and Computer Science
University of Göttingen
Göttingen, Germany
jeromario.schaefer@stud.uni-goettingen.de

Ulrich Sax
Department of Medical Informatics
University Medical Center Göttingen
Göttingen, Germany
ulrich.sax@med.uni-goettingen.de

Lena Wiese
Research Group Bioinformatics
Fraunhofer Institute for Toxikology and Experimental Medicine (ITEM)
Hannover, Germany
lena.wiese@item.fraunhofer.de

ABSTRACT

In this article we present the implementation and benchmarking of a medical information system on top of a distributed relational database system. We enhanced a distributed database system with the implementation of a clustering (based on similarity of disease terms) that induces a primary horizontal fragmentation of a data table and derived fragmentations of secondary tables. With our clustering-based fragmentation, data locality for similarity-based query answering is ensured so that data do not have to be sent unnecessarily over the network. In our benchmark we show that we achieve a significant efficiency gain when retrieving all relevant related answers.

CCS CONCEPTS

• **Information systems** → *Query optimization*; *Relational parallel and distributed DBMSs.*

KEYWORDS

Distributed database system, relational databases, similarity-based query answering

ACM Reference Format:
Jero Mario Schäfer, Ulrich Sax, and Lena Wiese. 2020. Benchmarking a Distributed Database Design that Supports Patient Cohort Identification. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3410566.3410608

1 INTRODUCTION

For the ever-growing amount of data in our world, *distributed databases* (DDBs) gained significantly more importance over the past years because they provide physically distributed storage [13, 16]. Especially in fields where the amount of data to be hosted in a database is challenging, a fragmentation of such big data as well as allocation to multiple servers in a cloud storage infrastructure can pay off. The data fragmentation overcomes the limitation of storage capacity of a single database server because the total amount of data can be split into smaller parts, so-called fragments. Moreover, fragmentation can increase the performance of query processing: queries of different users addressing different fragments that are stored on different servers can be answered independently and in parallel. Furthermore, the replication of the data fragments inside the network yields a tolerance to failures by compensating data loss with recovery, e.g. in case of a malfunction or total failure of a single server, to guarantee the desired availability and reliability of the data accessed by the users of the distributed database system (DDBS).

As a use case, the trend towards personalized medicine or the increased usage of novel biomedical technology – for example, next-generation sequencing [17] – produce vast amounts of data. In this article, our particular application scenario is a *medical information system* that uses a distributed database system as a storage backend. In our example system, patients' personal information as well as the diseases they suffer from are contained in a distributed database. A possible usage scenario is that researchers make use of these patient data, for example to identify a cohort of patients [1] that are similar to a current "target patient" – hence instead of exact query answering, a notion of similarity-based query answering is needed.

Our proposed data management system is supposed to support medical staff in identifying a relevant subset of patient data from partitioned tables in an efficient way. We make use of a taxonomy-induced data fragmentation and data distribution in a DDBS to achieve this. In order to support the use case of similarity-based query answering in an efficient way, we present here an implementation that enhances the basic distributed data management with an automatic clustering-based fragmentation that does not require any user interaction other than posing standard SQL queries.

A similarity measure defined between the disease terms from the MeSH taxonomy [9] yields the similarity values needed for the clustering. Our system ensures that data records holding information about patients suffering from similar illnesses are stored on the same site in a distributed database system; this enables the system to preserve data locality with respect to the semantics of the data as defined by the underlying taxonomy. In this way our system is able to answer similarity-based queries efficiently (without the need to access multiple servers).

Our specific contributions in this article are that we (1) obtain similarity values of disease terms by applying a shortest path algorithm in the Neo4J graph database; (2) implement a clustering

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410608

procedure on top of a relational database system that uses these similarity values; (3) compare the runtime behavior of similarity-based query answering with a round-robin approach and a clustering-based approach.

The remainder of this article is organized as follows. Section 2 surveys related work on flexible query answering and DDBs. Section 3 provides the necessary background on the employed database system. Section 4 introduces our notion of clustering-based fragmentation. Section 5 describes the implementation details and presents a comparative evaluation. Section 6 concludes the article.

2 RELATED WORK

2.1 Flexible Query Answering

Conventional database systems merely support exact query answering and are not supportive to the user when some query conditions cannot be satisfied. Upon query failure, that is, when the database cannot give an exact answer to a query formulated by a user, an empty result is returned. Assuming the query was formulated correctly, this empty result indicates to the user that the information he or she was looking for is not present in the current database instance. In this case of a lack of any exact answers, the empty database result is non-informative for the user in the general case. In contrast, real-life information systems should provide users with supportive mechanisms when they want to retrieve a particular information from the system. To wit, *flexible query answering* can overcome this lack of information when the database system uses techniques like *query relaxation* and *query generalization* to provide the user with a non-empty result. While this result does not match the query exactly, it nevertheless contains information instead that may be relevant to the user. This relevance property must be supported by an appropriate notion of similarity. A real-world use case is to find "patients like mine" in electronic health records [1].

Several approaches are based on different theoretical backgrounds for intelligent flexible query answering. A recent comprehensive survey of query relaxation in graph-structured data can be found in [12]. Opposed to this, [7] give an in-depth analysis of including taxonomic information on the relational algebra level. Several approaches consider similarity on the syntactical query level [5] or analyse combinations of so-called query generalization operators [3]. Furthermore, in contrast to the related work we explicitly consider a specific similarity in a taxonomy and propose a practical approach that considers query answering based on a semantic clustering. In addition, we devise a bridge between high-level similarity-based query answering and low-level distributed data management.

2.2 Distributed Databases

A *distributed database management system* (DDBMS) manages several underlying database instances and provides the access to the data spread across a computer network [11]. A partitioning of data tables can be achieved by using a certain fragmentation strategy and the resulting fragments, which contain parts of the whole data set, can then be dispersed across the network by mapping the fragments to database instances that possibly reside at different physical locations. A relational DDBS has to be able to answer a query in the same manner as a non-distributed relational database. The result of a query is a set of tuples fulfilling the query. To obtain the result, the database system uses relational algebra operations for its computation, e.g. a join of two tables or the selection on an attribute with a certain condition. As the data is physically distributed, queries have to be processed and rewritten according to the underlying distribution to allow for an appropriate answer to the given query. The performance of computing answers to the query can be a big issue due to increased network communication inferred by data transfer between the database sites. This can even occur for simple queries that only scan a certain relation and project to some subset of the attributes. Fragmentation and replication influence the query execution strategy as they require for a distribution of the query itself to possibly multiple servers, too, in order to get the complete and correct result set. In our system we apply appropriate rewriting techniques to support the intended similarity-based query answering.

2.3 Horizontal Fragmentation

One strategy to obtain a fragmentation of the data is *horizontal fragmentation* that divides a relation in a row-wise manner into smaller portions of tuples. More formally, a relation R is divided into fragments $F_1, F_2, ..., F_n$ by assigning each tuple μ of the relation R to at least one fragment The result of this is that for all $i \in \{1, \ldots, n\},\ F_i \subseteq R$. Additionally, in order to avoid redundancy of data, we can require that each tuple is only assigned to exactly one fragment; more formally, the fragments are pairwise disjoint: $\forall \mu \in F_i$ it holds that $\mu \notin F_j, i \neq j$ for $i, j \in \{1, \ldots, n\}$. In relational algebra such a *primary horizontal fragmentation* can be described by a selection operation σ on the relation R, where the selection condition defines the desired mapping of tuples to fragments. Based on this primary horizontal fragmentation, a further fragmentation of another relation S can be *derived* by computing the semi-join of the relation S with fragments $F_i,\ i \in \{1, \ldots, n\}$ of the primary relation R, i.e. the derived fragments G_i of the relation S are computed as $G_i = S \ltimes F_i$ for $i \in \{1, \ldots, n\}$. The *derived horizontal fragmentation* depends on the underlying primary horizontal fragmentation, and, to prevent tuples in S from getting lost during the semi-join with fragments of R, it is necessary to have for each tuple $y \in S$ matching tuples in R in order to let the tuples from S "survive" the semi-join. An integrity constraint in form of a foreign key reference of the relation S to the relation R can be used to enforce this condition for the sake of completeness of the derived fragmentation.

Several horizontal partitioning approaches in distributed database systems focus on numerical data. As an example for numerical data partitioning, the AdaptDB system [6] builds up a binary search tree where the median of a range of numerical values is chosen as the pivot element in each level of the search tree. Their storage model is based on low-level blocks of equal size in distributed file systems. Another approach [18] is predicate-based referenced partitioning (PREF) addressing relational DB systems. They start with a partitioning of a seed table and then co-partition all tables with incoming or outgoing foreign keys which could be potentially joined. This way of co-partitioning can lead to redundancy in the sense that tuples of the co-partioned tables can occur in more than one fragment. The co-partitionings are then further cascaded by

also co-partitioning tables that could be joined with the existing fragments.

Our approach differs from these two by supporting similarity-based query answering (which has a wide applicability in cohort identification on medical data) as well as enabling partitioning on categorical attributes (by clustering based on a similarity measure between terms). We visualize these differences in Table 1.

3 BACKGROUND

3.1 System Design

We will make use of horizontal partitioning to support similarity-based query answering. We postpone the formal definition until Section 4.3. We assume that one table is chosen as the primary table and one attribute of it is determined on which similarity-based query answering should be performed. Other tables that join to the primary table will be co-partitioned by derived fragmentation. Inherently with the definition of the derived horizontal fragmentation on a semi-join, redundancy of tuples of S in the derived fragments may occur if tuples in S match multiple tuples that belong to different fragments of R. This causes the fragments G_i of S to be non-disjoint in general – an approach also followed in [18]. We will make use of this property to ensure data locality of primary and derived fragments: while we require the primary fragmentation to be disjoint (and hence non-redundant) the derived fragmentation might contain fragments that have some tuples in common. These derived fragments are however stored on different sites together with their matching primary fragment.

3.2 Default Hash Partitioning

Our specific implementation reported on in this paper is based on in-memory storage inside a network of servers each running an Apache Ignite instance. The Ignite server nodes store data depending on the fragmentation (called partitioning with Ignite) and replication. The fragmentation and replication is defined per relation. The default Ignite partitioning is a horizontal fragmentation of the data defined with hash functions; thus, there are no explicit selection conditions on a certain attribute of the relation that represent a horizontal fragment as we would need for our semantic clustering-based fragmentation – and the assignment of tuples to the partitions/fragments is done according to the hash function and hence rather arbitrarily.

3.3 Ignite's Affinity Collocation

One important concept of Ignite is the collocation of data – which corresponds to the concept of a derived horizontal fragmentation: data that are accessed together, e.g. because they are joined via a common attribute or a foreign key reference, are also stored together on the same server. This affinity collocation can be defined in Ignite by so-called *affinity keys*: An affinity key can be identical to the primary key of a relation or an attribute of a composite primary key of a relation. Ignite ensures that all tuples where the affinity keys match are stored on the same server. The usage is restricted to a single affinity key definition per relation. With this restriction, there cannot be a collocation of three or more relations that could be joined via a chain of join conditions where the join attributes would form possible affinity keys of the different relations. Note that this notion of affinity for horizontal fragmentation is different from the notion of attribute affinity which has long since been used for vertical fragmentation [10].

If all the joined relations in the SQL query are collocated, the query can be evaluated locally by each node, because all the data they need to compute a correct result set regarding their portion of the whole data in the cluster is available, i.e. stored by themselves. Non-collocated joins must be enabled explicitly in Ignite to enforce the distributed answering with data transfer across the network if necessary. If not enabled explicitly, the query will be executed only in a collocated (local) manner which, in general, leads to incomplete result sets due to required data not being available locally.

4 CLUSTERING-BASED FRAGMENTATION

To improve data collocation from a semantic point of view, we replace the hash-based horizontal partitioning by our proposed clustering-based fragmentation. More precisely, the clustering-based fragmentation is a horizontal fragmentation strategy that, on the one hand, enables fragmentation regarding a similarity measure which allows for a semantic partitioning of the data set, and, on the other hand, supports similarity-based query answering.

4.1 Similarity Computation

A specific measure of similarity has to be defined on terms connected in a taxonomy (or more generally any ontology on which we can define a notion of similarity). The pairwise similarity values between any two terms in the taxonomy form the basis for a clustering on the terms contained in one of the attributes in the provided database tables. More formally, we assume that in our primary relation R there is a specific attribute A which will be used for clustering and hence similarity-based query answering. In order to capture semantic closeness, we need a similarity relationship $sim(a, b)$ for pairs of elements a, b from the *active domain* of the chosen attribute A: the projection $\pi_A(R)$ of R to the values of A. More formally, $sim : \pi_A(R) \times \pi_A(R) \rightarrow \mathbb{R}$. It is customary to restrict the range of the similarity to the interval $[0, 1]$ such that a similarity of 1 denotes that the elements a and b have highest similarity, whereas the closer the similarity value gets to 0, the more dissimilar the two elements are. In cases where the terms occurring in the queries are not contained in the active domain, we have to obtain the similarity for each such query term to the terms of the active domain, too. Hence we assume that all terms in the active domain of the selected attribute as well as in any value for the attribute required in a query are contained in the taxonomy (or at least can be mapped to a term in the taxonomy).

In our application of a medical information system, a similarity is defined on the disease information of patients; we assume that the disease terms conform to the vocabulary provided by the *Medical Subject Headings* taxonomy (MeSH) of the U.S. National Library of Medicine [9]. In MeSH the same disease term can be located under different subtrees (corresponding to different disease classifications). The levels in the classification tree are represented by numbers; so each term can be uniquely identified by its "tree number".

We imported the MeSH taxonomy into the graph database Neo4J. We used Neo4J because it has a convenient query language called Cypher as well as provides a graph algorithms library that can be

	primary partitioning method	secondary partitioning method	query method	storage
AdaptDB [6]	median of the attribute in root of partitioning tree	medians of other attributes in the partitioning tree	exact	block level (HDFS)
PREF [18]	hash-based	co-partitioning on join attributes	exact	relational (XDB)
Our approach	clustering on active domain of an attribute	co-partitioning on join attributes	similarity-based	relational (Apache Ignite)

Table 1: Comparison of approaches

Figure 1: A snippet of the MeSH taxonomy in Neo4J

used to compute shortest paths. As shown in Table 2 we read in the MeSH entries from a CSV file where the first entry in each line is the tree number and second entry is the disease term. First of all we create a node for every term in the file resulting in 11,649 nodes. Next we create edges between a term and its parent term by identifying the parent term by its tree number (which is the tree number of the term without the last 4-digit level) resulting in 11,648 edges. Figure 1 shows a snippet of the obtained tree where the Diseases category is the root node.

Next, we obtain the minimal length shortest path between two disease terms by a Cypher query that looks up any two disease terms, using the Neo4J graph algorithms library to obtain the shortest paths and returning the length of the shortest one (that is, the amount of edges on the path). Lastly, we obtain a similarity value by dividing 1 by the shortest path length plus 1 (that is, the amount of nodes on the shortest path):

$$sim(a,b) = \frac{1}{minlength(shortestpath(a,b)) + 1} \quad (1)$$

This approach of similarity defined by shortest path length can generally be applied to any graph-shaped ontology.

We could of course compute any similarity value on demand by issuing a query to Neo4J. However we need all pairwise similarities for the clustering up front. Hence, for sake of performance we precompute all similarity values and store them in a similarity table in the Ignite system that and similarity values are read from this table while performing the clustering.

4.2 Clustering

When loading the data into the distributed database, the underlying clustering is computed with an approximation algorithm [2] on all values that occur in the active domain of a chosen attribute. The pseudocode of the clustering procedure is described in Listing 1. The clustering starts with a single cluster (Line 1) containing the whole active domain and an arbitrarily chosen representative "head" element from the cluster and then identifies the minimal similarity inside the cluster between all elements from the active domain and the cluster head (Line 6). Subsequently, new clusters are created with new head elements based on the minimal similarity of a term to the head of a cluster as long as the similarity threshold is not exceeded; all elements are reassigned if they are more similar to the head of the newly created cluster (Lines 7 and 8). The procedure iterates as long as there are still elements inside one of the clusters that have a similarity to the corresponding head element that is lower than a user-defined similarity threshold α (while-condition in Line 5). Hence, the iteration proceeds until each element of the active domain is clustered such that the minimal similarity according to threshold α can be ensured.

Listing 1 Clustering procedure

Input: Set $\pi_A(F)$ of values for attribute A, similarity threshold α
Output: A set of clusters $c_1, \ldots, c_f$

1: Let $c_1 = \pi_A(F)$
2: Choose arbitrary $head_1 \in c_1$
3: $sim_{min} = min\{sim(a, head_1) \mid a \in c_1; a \neq head_1\}$
4: $i = 1$
5: **while** $sim_{min} < \alpha$ **do**
6: Choose $head_{i+1} \in \{b \mid b \in c_j; b \neq head_j; sim(b, head_j) = sim_{min}; 1 \leq j \leq i\}$
7: $c_{i+1}=\{head_{i+1}\} \cup \{c \mid c \in c_j; c \neq head_j; sim(c, head_j) \leq sim(c, head_{i+1}); 1 \leq j \leq i\}$
8: $c_i=c_i \setminus \{c \mid c \in c_j; c \neq head_j; sim(c, head_j) \leq sim(c, head_{i+1}); 1 \leq j \leq i\}$
9: $i = i + 1$
10: $sim_{min} = min\{sim(d, head_j) \mid d \in c_j; d \neq head_j; 1 \leq j \leq i\}$
11: **end while**

As an example, consider the sample disease terms *Hemoptysis*, *Acute-On-Chronic Liver Failure*, *Massive Hepatic Necrosis*, *Hepatic Encephalopathy*, and *Ulna Fracture* from Figure 1 as the active domain of the diagnosis attribute. We see that all liver diseases are similar to one another (either $\frac{1}{3}$ or $\frac{1}{4}$) – but are less similar to lung diseases (either $\frac{1}{9}$ or $\frac{1}{10}$), and vice versa. By, for example, setting α

Create nodes	`USING PERIODIC COMMIT LOAD CSV FROM "file:///ctree2019MeSH.csv" AS line` `CREATE (d:Disease {TREE_NUMBER:toString(line[0]),DESCRIPTOR:toString(line[1])});`

Create edges	`USING PERIODIC COMMIT LOAD CSV FROM "file:///ctree2019MeSH.csv" AS line`
	`MATCH (d:Disease {TREE_NUMBER: toString(line[0])}) WHERE size(d.TREE_NUMBER)>4`
	`MATCH (d2:Disease {TREE_NUMBER: substring(d.TREE_NUMBER,0,(size(d.TREE_NUMBER)-4))})`
	`MERGE (d)-[:PARENT]->(d2)`

Return shortest path length	`MATCH (n:Disease {DESCRIPTOR: "Massive Hepatic Necrosis"})`
	`WITH n MATCH (n2:Disease {DESCRIPTOR: "Hemoptysis"})`
	`WITH n,n2 MATCH p = shortestPath((n)-[*]-(n2)) RETURN min(length(p))`

Table 2: Loading MeSH into Neo4J and finding shortest paths

between $\frac{1}{5}$ and $\frac{1}{8}$ we can separate liver diseases and lung diseases into two separate clusters.

The different clusters contained in the clustering are then used to obtain a horizontal fragmentation of the entire relation instance R by partitioning the relation along the disjunctive subsets of elements from the active domain with respect to the chosen attribute A. Each cluster induces one horizontal fragment: any two tuples are mapped to the same fragment if their values for clustering attribute A belong to the same cluster.

Completeness of the clustering provides coverage for all elements of the active domain $\pi_A(R)$ regarding the attribute A. Furthermore, the mapping of any element of the active domain to one of the clusters is functional. This makes the clusters pairwise disjoint, and hence, all fragments induced by the clustering are non-redundant; each tuple is assigned to only fragment according to the maximum similarity to one of the cluster heads. Thus, the relation instance R can be reconstructed from the horizontal fragments.

4.3 Query Answering

As the data is distributed according to the clustering into horizontal fragments, similarity-based queries can now be executed in a distributed manner according to the clustering. Finding the relevant data fragment is done based on the selection condition on the attribute A chosen for the clustering: for each term in the selection condition the term's similarity to the head elements of the different clusters is obtained. The maximal similarity of the comparison element and all cluster heads determines the matching cluster as well as the induced horizontal fragment. More formally, we can define a similarity-based answer for each selection (sub-)query on attribute A as follows. Given a clustering-based fragmentation of relation R, for a selection query $\sigma_{A=\text{“}s\text{”}}R$ on the clustering attribute A the similarity-based answer is

$$\{F_i \mid head_i = argmax_{j=1,\ldots,n} sim(s, head_j)\}$$

If there is more than one cluster head with maximal similarity, we choose one of them at random. In this case it is sufficient to execute the query only locally at a single server which hosts the single relevant data fragment. An alternative to this approach would be to return the union of all fragments with most similar head elements.

5 IMPLEMENTATION AND EVALUATION

To evaluate the proposed clustering-based fragmentation, we compare its implementation to the default hash-based partitioning. The source code is available in a Github repository[1].

5.1 Data Set

We generated a synthetic data set to comparatively investigate the behavior of the implementation variants. We analyzed scalability of our approach by varying both the data set size and the size of the term set (that is, active domain of the clustering attribute) on which similarities are calculated. Our test data set is modelled according to three tables: A table “Ill” (containing attributes for the patient ID and the diagnosis), a table “Treat” (containing attributes for the patient ID and the medication) and a table “Info” (containing attributes for the patient ID and additional administrative data like address and age). In this way we simulate a simplified database schema found in medical datasets like MIMIC [4] or platforms like i2b2 [8].

The Ill table is our primary table; the patient IDs are generated in decreasing order; the diagnosis attribute contains disease terms extracted randomly from the MeSH data set. The Treat table contains the patient IDs and randomly generated string data as the prescription. The Info table contains one random address string for each patient as well as a randomly generated age.

Scaling of the data set was obtained by a default number of tuples for each of the tables multiplied by a scaling factor. The default size of the Ill table is 100 tuples and the default size of both Treat and Info table is 50 tuples – that is, an average of two disease entries per person plus one sample prescription. For a given scaling factor s the dataset is expanded to a total of $s \cdot (100+50+50) = 200 \cdot s$ tuples. The MeSH term set was divided into smaller, randomly chosen subsets ranging from a minimum of 100 terms up to all 4798 terms from which the Diagnosis column of the Ill table is filled.

[1] https://github.com/l-wiese/SiFAMIS

Q_1	`SELECT p.name, p.age, p.address FROM ILL i, INFO p WHERE i.id = p.id AND i.disease='Hepatic Encephalopathy'`
Q_2	`SELECT p.name, p.age, p.address FROM ILL i1, ILL i2, INFO p WHERE i1.id = p.id AND i2.id = p.id AND i1.disease='Hepatic Encephalopathy' AND i2.disease='Hemoptysis'`
Q_3	`SELECT t.prescription FROM ILL i,TREAT t WHERE t.id = i.id AND i.disease = 'Hepatic Encephalopathy'`
Q_4	`SELECT p.name, p.age, t.prescription FROM ILL i,TREAT t, INFO p WHERE t.id = i.id AND i.id = p.id AND i.disease = 'Hepatic Encephalopathy'`
Q_5	`SELECT t.prescription FROM ILL i, ILL i2, TREAT t WHERE i.id = i2.id AND i.id = t.id AND i.disease = 'Massive Hepatic Necrosis' AND i2.disease = 'Hepatic Encephalopathy'`

Table 3: Benchmark queries (disease terms are chosen randomly)

5.2 Clustering-Based Implementation

Before inserting data, we have a certain one-time overhead in terms of initialising the similarity table inside Ignite and clustering the active domain of the selected attribute:

- For clustering the disease attribute (implemented as an external Java program) we measured runtimes of 0.96 seconds for 500 MeSH terms, 2.05 seconds for 1000 MeSH terms, 13.60 seconds for 2500 MeSH terms, 75.61 seconds for all 4798 MeSH terms.
- For the initialisation of the similarity table we measured runtimes of 8.21 seconds for 500 MeSH terms, 28.82 seconds for 1000 MeSH terms, 109.00 seconds for 2500 MeSH terms, 228.03 seconds for all 4798 MeSH terms.
- For the batch loading of our data set for 200000 tuples (that is, scale factor 100) which includes a lookup on the similarity table we measured runtimes of 88.57 seconds for 500 MeSH terms, 115.45 seconds for 1000 MeSH terms, 162.48 seconds for 2500 MeSH terms, 258.66 seconds for all 4798 MeSH terms.

Yet we assume that this kind of batch loading of a large data set only occurs once before actually using the system. Hence the performance gains during querying (as shown later in Section 5.5) pay off when querying the system. As discussed in Section 6 modifications of the data set are up to future work.

In order to deploy the clustering-based approach with Ignite, we assign different partition numbers (each corresponding to one cluster) to the different semantic fragments of the relations via Ignite's affinity function API. The fragments are then distributed and mapped to the servers. Collocation of derived partitions is also achieved with the help of the affinity function and the identification of the correct partition number that is inferred from the clustering.

In our data set we can join both the *Treat* as well as the *Info* table with the *Ill* table based on the patient ID. For any fragment of *Ill*, we obtain one derived fragment of each of *Treat* and *Info*: we fragment the *Treat* as well as the *Info* table according the patient IDs contained in the fragments of the primary table *Ill*. For example, if a patient has disease x that belongs to cluster c_j, then partition j of the *Ill* relation stores the tuple stating that this patient has disease x and additionally partition j of the *Info* relation is responsible for the tuple with the patient's personal information.

In order to implement similarity-based query answering, the selection condition on the clustering attribute is omitted and the query is adapted by restricting it to the fragment that belongs to the cluster with the relevant diseases, such that then all answers only need to be obtained from this fragment; the partition number has to be identified and set via the appropriate class method of the SqlQuery or SqlFieldsQuery of Ignite's SQL API.

5.3 Default Implementation

On the other hand, the default hash-based partitioning is implemented by creating partitioned tables by the standard Ignite partitioning methodology. The primary table (in our example, *Ill*) is partitioned horizontally based on a hash function applied to its affinity key (that is, attribute patient ID); The other tables are collocated via their shared attribute, the patient ID, such that personal information and the diseases and treatments of a patient are stored together to ensure the collocation of the data. That is, only collocation via the attribute patient ID is guaranteed – but no similar disease terms are collocated. In order to implement similarity-based query answering, the selection condition on the clustering attribute is replaced by a more general expression: the original SQL query is translated into one with a SQL `IN`-clause containing the similar disease terms from the appropriate cluster.

5.4 Queries

Our benchmark queries Q_1 to Q_5 (see Table 3) consist of executing joins (between primary and secondary fragments) with a selection condition on the diagnosis attribute. Similarity-based query execution includes finding the fragment that is closest to the query condition (in terms of similarity to the cluster head that is contained in the diagnosis attribute of the fragment). In other words, query execution extracts the selection condition, applies the query rewriting and returns the obtained fragment (or joins of fragments, respectively) as the set of related answers. Note that queries Q2 and Q5 both have two selection conditions on which similarity-based query-answering is applied. The difference between the two queries is that in Q5 both selection conditions come from the same fragment (and the tuples are hence collocated); whereas in Q2 the selection conditions are not in the same cluster and data have to be retrieved from different fragments before joining them.

5.5 Results

Our distributed system is evaluated in a network of three Apache Ignite nodes where each of the nodes runs in a JVM and is hosted by one of three servers. A total of 24 GB memory for the cloud is split equally among all machines and each server has 4 processors.

Table 4 shows the results of executing our five benchmark queries (Table 3) when scaling the amount of tuples in the database from 20000 over 200000 to 2000000. For the clustering-based approach we tested in addition different amounts of underlying disease terms

Query	# Tuples	Default	Partitions 100 terms	Speed up	Partitions 500 terms	Speed up	Partitions 1000 terms	Speed up	Partitions 2500 terms	Speed up	Partitions all terms	Speed up
1	20000	171.57	80.56	2.13	59.15	2.90	71.09	2.41	59.11	2.90	61.70	2.78
2	20000	269.27	112.30	2.40	58.30	4.62	170.10	1.58	121.51	2.22	102.83	2.62
3	20000	180.99	91.21	1.98	55.20	3.28	108.45	1.67	82.77	2.19	86.56	2.09
4	20000	108.41	113.47	0.96	56.56	1.92	89.41	1.21	76.68	1.41	64.20	1.69
5	20000	188.60	92.11	2.05	55.80	3.38	136.02	1.39	87.87	2.15	89.94	2.10
1	200000	424.51	148.24	2.86	196.29	2.16	135.11	3.14	107.74	3.94	125.16	3.39
2	200000	4000.00	199.12	20.09	157.96	25.32	146.65	27.28	382.47	10.46	327.74	12.20
3	200000	357.09	138.97	2.57	147.80	2.42	260.87	1.37	534.77	0.67	385.51	0.93
4	200000	1362.86	146.70	9.29	132.54	10.28	262.20	5.20	250.47	5.44	309.57	4.40
5	200000	4000.00	181.84	22.00	181.10	22.09	275.14	14.54	256.14	15.62	289.77	13.80
1	2000000	4000.00	1168.24	3.42	1314.28	3.04	1150.21	3.48	1257.63	3.18	1160.64	3.45
2	2000000	4000.00	2410.04	1.66	2009.41	1.99	2301.47	1.74	2391.26	1.67	2271.32	1.76
3	2000000	4000.00	1120.11	3.57	1350.73	2.96	1540.83	2.60	1747.32	2.29	1670.13	2.40
4	2000000	4000.00	1296.70	3.08	1396.45	2.86	1320.13	3.03	1520.87	2.63	1309.04	3.06
5	2000000	4000.00	2411.80	1.66	2566.07	1.56	2700.19	1.48	2619.55	1.53	2729.25	1.47

Table 4: Runtime measurements for queries 1 to 5 in milliseconds for varying amount of tuples in the database instance and varying amount of MeSH terms in the active domain of the clustering attribute; the default approach is stopped when exceeding 4 seconds; speedup is relative to the default approach; overall average speedup of our approach is 4.79

from MeSH that are used in the Diagnosis column: we tested 100, 500, 1000, and 2500 randomly chosen as well as all (4798) MeSH terms. We ran the five queries three times and averaged the runtimes. Whenever the default Ignite approach runtime significantly exceeded the runtime of the other approaches, we cut off its measurements after 4000 milliseconds. Our measurements show that the clustering-based approach scales better for the similarity-based query answering use case. For the largest scaling factor (2000000 tuples) the default approach always faced a timeout whereas the average query execution time of our similarity-based approaches was 1530 milliseconds. For most of the cases the similarity-based approach also performs better for smaller data sets (smaller scaling factors); only for some term set sizes in Queries 3 and 4, the similarity-based query answering introduces a slight overhead – evidenced by a speed up < 1 compared to the default Ignite approach. Averaged over all measurements we achieve a speedup of 4.79

6 CONCLUSION

The presented distributed database design demonstrates the capabilities of our novel similarity-based query answering where data fragmentation is based on a clustering with respect to a given similarity. The query execution runtimes show that the clustering-based fragmentation improves the execution time of queries against the DDB significantly when comparing to the basic implementation that provides only an arbitrary, hash-based horizontal fragmentation of the data.

In future work, other taxonomies and other disease similarities could be used – depending on the analyzed medical use case. In addition, several notions of similarity (semantic as well as numeric as in [14, 15]) can be combined in order to identify not only patients that suffer from similar diseases but also whole patient profiles based on the similarity of their personal characteristics (e.g. their age or weight) and some other recorded measurements (e.g. body temperature or blood parameters). Modifications in the data set are crucial for a real-world applications. In future work we plan to support adaptivity to changing attributes values (by either insertions, deletions or updates) and hence changing clusters. Further practical advancements include analysis of other clustering methods and their influence on the resulting distributed data management behavior.

ACKNOWLEDGEMENTS

This work was partially supported by the Fraunhofer Internal Programs under Grant No. Attract 042-601000.

REFERENCES

[1] S. Gombar, A. Callahan, R. Califf, R. Harrington, and N. H. Shah. It is time to learn from patients like mine. *npj Digital Medicine*, 2(1):1–3, 2019.
[2] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293 – 306, 1985.
[3] K. Inoue and L. Wiese. Generalizing conjunctive queries for informative answers. In *International Conference on Flexible Query Answering Systems*, pages 1–12. Springer, 2011.
[4] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.
[5] V. Kantere. Query similarity for approximate query answering. In *International Conference on Database and Expert Systems Applications*, pages 355–367. Springer, 2016.
[6] Y. Lu, A. Shanbhag, A. Jindal, and S. Madden. Adaptdb: adaptive partitioning for distributed joins. *Proceedings of the VLDB Endowment*, 10(5):589–600, 2017.
[7] D. Martinenghi and R. Torlone. Taxonomy-based relaxation of query answering in relational databases. *The VLDB Journal*, 23(5):747–769, 2014.
[8] S. Murphy and A. Wilcox. Mission and sustainability of informatics for integrating biology and the bedside (i2b2). *eGEMs*, 2(2), 2014.
[9] National Library of Medicine. Medical subject headings, Nov 2019.
[10] S. Navathe, S. Ceri, G. Wiederhold, and J. Dou. Vertical partitioning algorithms for database design. *ACM Transactions on Database Systems (TODS)*, 9(4):680–710, 1984.
[11] M. T. Özsu and P. Valduriez. *Principles of Distributed Database Systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1991.
[12] A. Poulovassilis. Applications of flexible querying to graph data. In *Graph Data Management*, Data-Centric Systems and Applications, pages 97–142. Springer, 2018.

[13] K. Tan. Distributed database systems. In *Encyclopedia of Database Systems (2nd ed.)*. Springer, 2018.
[14] A. Tashkandi, I. Wiese, and L. Wiese. Efficient in-database patient similarity analysis for personalized medical decision support systems. *Big data research*, 13:52–64, 2018.
[15] I. Wiese, N. Sarna, L. Wiese, A. Tashkandi, and U. Sax. Concept acquisition and improved in-database similarity analysis for medical data. *Distributed and Parallel Databases*, pages 1–25, 2018.
[16] L. Wiese. *Advanced Data Management for SQL, NoSQL, Cloud and Distributed Databases*. DeGruyter/Oldenbourg, 2015.
[17] L. Wiese, A. O. Schmitt, and M. Gültas. Big data technologies for DNA sequencing. In *Encyclopedia of Big Data Technologies*. Springer, 2019.
[18] E. Zamanian, C. Binnig, and A. Salama. Locality-aware partitioning in parallel database systems. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 17–30, 2015.

Lifting Preferences to the Semantic Web: PreferenceSPARQL

Markus Endres, Stefan Schödel, Klaus Emathinger
University of Passau
Passau, Germany
markus.endres@uni-passau.de,klaus.emathinger@uni-passau.de

ABSTRACT

PreferenceSQL is an SQL extension for standard relational databases supporting soft constraints and is used to find relevant data intuitively. Meanwhile, the Semantic Web has interoperability advantages and helps to retrieve information with machine-readable data. We use the benefits of both technologies by combining preferences from SQL with SPARQL, the query language of the Semantic Web. This work provides implementation details in Apache Jena for the new composite called 'PreferenceSPARQL'. Furthermore, we contribute comprehensive benchmarks that show which preference algorithm is best suited for our approach.

CCS CONCEPTS

• **Information systems → Query languages for non-relational engines**;

KEYWORDS

Information systems, Data management systems, Query languages

ACM Reference Format:
Markus Endres, Stefan Schödel, Klaus Emathinger. 2020. Lifting Preferences to the Semantic Web: PreferenceSPARQL. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3410566.3410590

1 INTRODUCTION

The primary audience of the World Wide Web is human users. The data is unstructured, mostly represented as free-form text, and the organization principles are weak with different kinds of information co-existing. These factors make it unsuitable for machine consumption [15].

Instead of just serving as a passive display of information, the vision of the Semantic Web is an intelligent system capable of assisting humans in the creation of meaning [17]. Information is modelled, manipulated and queried at the conceptual level [15].

One of the building blocks for the Semantic Web is the Resource Description Framework (RDF), a data model and language for describing web resources. The SPARQL Protocol and RDF Query Language (SPARQL) is the de facto standard for querying RDF data. It realizes the filtering of information via hard constraints. Either a binding satisfies a filter expression and is added to the output or it does not, leading to the binding's exclusion from the output. However, a user expects a reasonable amount of results to be delivered. Hence, the query is manually adjusted until a reasonable result set is returned.

To facilitate an efficient and effective exchange of information, the filtering of such information must also evolve. If relevant data is sparse, an intelligent filter will relax its constraints to present the next best options. If relevant data is abundant, only the best results as inferred by some implicit ranking should be returned.

Preferences are a solution for intelligent filtering [14]. Users state their preferences by adding soft constraints which more faithfully reflect the underlying intention of the user [8, 12]. Preferences can be interpreted as personalized wishes in the form of *'I like A more than B'* that are formalized by the strict partial order preference model [9]. Built upon this theoretical framework, PreferenceSQL[1] [10] extends the SQL standard introducing preferences to relational databases. Following a constructor-based approach, preference queries can be formulated intuitively and support a multi-criteria decision by default.

The objective of this paper is the seamless expression of user preferences in the Semantic Web. In addition, we want to provide comprehensive benchmarks w.r.t. different preference algorithms. For this, we developed and evaluated a SPARQL extension called 'PreferenceSPARQL', which supports strict partial order preferences natively, as shown in the example.

Example 1. *Assume we want to purchase an apartment and we already have a specific size in mind (e.g., 75 square metres). More important is the price, which is deemed affordable (let's say at most 250 000 Euros). Accordingly, in Figure 1 a simplified PreferenceSPARQL query is shown expressing our wishes in the 'prefer' clause:*

```
prefix : <http://example.com/real-estate/>

select ?sale_offer ?price, ?size where
{
  ?sale offer a : sale_offer ;
                : price_eur ?price ;
                : size_sqm ?size ;
  prefer (
        ?size around 75 prior to
        ?price lower than 250000)
}
```

Figure 1: A simple PreferenceSPARQL query.

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410590

[1] PreferenceSQL: http://www.preferencesql.com

In this work, we developed the powerful and flexible PreferenceSPARQL framework, which includes the major base and complex preference constructors presented in [10]. Input data may be partitioned (grouped) along multiple axes, while the preference selection is to be computed for each partition (group) separately. Finally, preference query execution is to be delegated to a varied range of independent preference algorithms: Block-Nested Loop (BNL), Sort and Limit Skyline algorithm (SaLSa), Linear Elimination Sort for Skyline (LESS) and Query Rewriting.

The rest of the paper is organized as follows: In Section 2 we discuss some related work. Section 3 describes the foundations of preferences and introduces concepts of the Semantic Web. In Section 4 we describe the new composite PreferenceSPARQL. Section 5 shows the setup and results of our comprehensive experiments. In the final Section 6, we summarize the most important aspects of this paper, discuss limitations, and give an outlook for further research.

2 RELATED WORK

An early prototype of preferences in the Semantic Web was done by Siberski et al. [13]. They implemented a preference query formalization of Chomicki [4] into ARQ, the Apache Jena SPARQL engine. They delegated the preference evaluation, which was restricted to HIGHEST (maximum), LOWEST (minimum), Pareto (equal important preferences), and Prioritization (more importance), to BNL. Further limitations of this approach are a lack of notion for regular SV semantics (cp. [10]), no partitioning values (without relying on an embedding into the boolean expression) and the absence of any evaluation. Pivert et al. [12] published a good survey on SPARQL extensions with preferences that are classified into quantitative and qualitative ones, but they did neither discuss implementation concepts nor presented experiments. Gueroussova et al. [7] have developed a qualitative approach that adds a preferring graph pattern and rewrites queries to SPARQL. It supports multiple atomic constructors and conditional preferences in the form If E Then P_1 Else P_2.

Newer papers, e.g., [16], discuss qualitative preferences, presenting SPREFQL, an extension of the SPARQL language that allows appending a "PREFER" clause to express soft constraints. This work is close to ours, but only allows Pareto and Prioritization preferences and only presents experiments on a view simple preference queries using the standard BNL algorithm. Patel-Schneider et al. [11] focus on comparative preferences in SPARQL, extended the syntax of [16], and repaired a problem identified with the translation of preference queries into SPARQL found in [16]. However, they do no show any query evaluation.

We use an extension of those approaches and alternative execution strategies in our comprehensive benchmarks. In addition, we solved all these drawbacks and extended it with numerous base and boolean preferences to define quantitative and qualitative preferences.

3 BACKGROUND

3.1 Preferences

Preference queries are grounded in the observation that users easily describe their desires in sentences akin to *'I like Y more than X'*. We follow the preference model developed by [10]. Formally, we can write $X <_P Y$, where $<_P$ is a strict partial order (SPO). P denotes to a preference on a set of attributes A and is defined as $P := (A, <_P)$, where $<_P \subseteq \text{dom}(A) \times \text{dom}(A)$. A base constructor operates on a single attribute of either categorical or numerical domain. Figure 2 schematizes the hierarchy of the major base constructors. Each edge indicates a subsumption relationship.

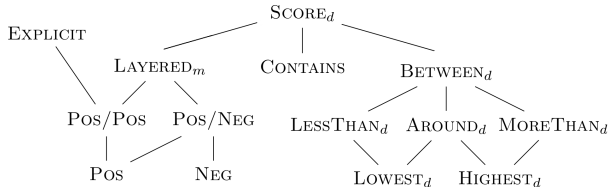


Figure 2: Taxonomy of base preference constructors.

For the SCORE$_d$ preference a scoring function $f : \text{dom}(A) \to \mathbb{R}$ with a discretization factor $d \geq 0$ is used. A preference P is called a SCORE$_d(A, f)$ preference iff $\forall x, y \in \text{dom}(A)$: $x <_P y \iff f_d(x) > f_d(y)$. The discretization factor d divides continuous values into classes of equal importance.

BETWEEN$_d$, a constructor for the numerical domain stating the desire for a value between a lower and an upper bound. Within the threshold $f(v) = 0$ applies, below $f(v) = low - v$ and above $f(v) = v - up$. Other numerical preferences are a specialization of BETWEEN$_d$. For example by setting low = up, we arrive at AROUND$_d(A, z)$. Categorical base preferences are sub-constructors of LAYERED$_m$. Let $m \geq 0$ and $L = (L_1, \ldots, L_{m+1})$ be an ordered list of $m + 1$ disjoint sets of dom(A). Then a LAYERED$_m(A, L)$ preference is a SCORE preference with the subsequent utility function: $f(x) := i - 1 \iff x \in L_i$. Complex constructors combine multiple preferences into a single preference. For example, a Pareto preference treats two preferences ($P := P_1 \otimes P_2$) equally, while a Prioritization ($P := P_1 \& P_2$) ranks them in sequential order. An example of Prioritization can be seen in Figure 1.

Preference selection performs intelligent filtering by taking the quality of the available data and the stated wishes of the user into account. The result of preference selection is a Best-Matches-Only-set (BMO-set). In PreferenceSQL [10] the discussed constructors are part of the PREFERRING clause. Additionally, there is a GROUPING clause that allows partitioning with an attribute list. P is then evaluated on each partition separately. Lastly, BUT ONLY can be used for filtering in a similar way to WHERE only that it is used after P was evaluated.

3.2 Semantic Web and RDF

The Resource Description Framework (RDF) represents information via a set of statements which can be visualized as a labelled, directed graph. Each RDF statement consists of a triple in the form $\langle subject, predicate, object \rangle$, where we can distinguish three different types: resources, literals and blank nodes. A resource is described by a Uniform Resource Identifier (URI) within a global namespace [18]. Literals encode necessary information like numbers, dates and strings. While blank nodes provide a way of introducing resources without explicitly naming them. RDF graphs

are obtained by combining multiple triples, as shown in the next example.

Example 2. *Figure 3 shows a small RDF graph describing the chemical element 'Titanium'. Each edge starts at the subject, is labelled with the predicate and points towards the object. A rectangle is used for literals and an oval for resources. It is visible that the object of one triple can be the subject of another triple. It proposes a graph-structural data model. We obtain an RDF data set by collecting several such graphs.*

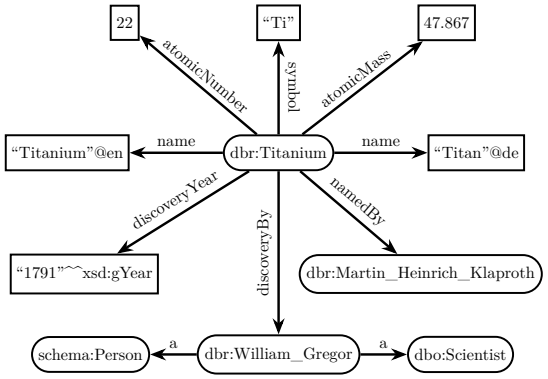


Figure 3: Sample RDF graph of the chemical element Titanium.

The goal of RDF is to connect heterogeneously structured data from multiple sources [5]. The SPARQL Protocol and RDF Query Language support entailment regimes for RDF Schema, Web Ontology Language (OWL), and others. A SPARQL SELECT query usually starts with prefix and base statements. These are responsible for declaring URI abbreviations and defining a base for relative URIs, respectively. A SELECT clause follows after that and defines a list of variables that is projected. The WHERE clause defines the query graph pattern. It can contain several basic graph patterns (triples) that can be combined with conjunctions and disjunctions. Another part of the WHERE clause is filters that restrict the solution sequence S to those solutions that satisfy a constraint R, written as S FILTER R. A SPARQL query (extended by preferences) is shown in Figure 1. Similar to SQL, SPARQL also supports grouping via GROUP BY, sorting via ORDER BY and LIMIT to restrict the result size. There are even more clauses, but only SELECT and WHERE are mandatory (the WHERE keyword may be elided).

4 PREFERENCESPARQL

We now introduce 'PreferenceSPARQL' as a query language for RDF data with preferences and discuss some implementation details in Apache Jena.

4.1 Query Language

The goal of PreferenceSPARQL is to incorporate the advances made with PreferenceSQL [10] into SPARQL. This includes the major base and complex preference constructors described above. Furthermore, a partitioning mechanism similar to GROUPING called 'PARTITION' has been realized. In this sense, we follow the syntax presented in [16] and [11], but provide comprehensive experiments in Section 5.

In order to be a first-class citizen of the SPARQL, our extension point should be a graph pattern. The SPARQL Filter is a prime candidate for imitation. An alternative approach would be the implementation of a solution modifier. This option, however, comes with the drawback of needlessly complicating queries with nesting (subqueries) if multiple independent preference statements are desired, e.g., for human readability. Given the preceding rationale, we have imitated the SPARQL Filter, i.e., the SPARQL grammar has been extended with a PREFER clause on the following rule:

```
GraphPatternNotTriples :==
  | GroupOrUnionGraphPattern
  | OptionalGraphPattern | MinusGraphPattern
  | GraphGraphPattern | ServiceGraphPattern
  | Filter | Bind | InlineData
  | Prefer
```

In order to define PREFER and PARTITION, let P be a graph pattern, *Pref* be an inductively constructed preference and G be a set of variables. Then the term P PREFER *Pref* PARTITION G is a graph pattern. We write P PREFER *Pref* as shorthand for partitioning by an empty set G. PREFER and PARTITION realize PreferenceSQL's PREFERRING and GROUPING, respectively. A BUT ONLY can be emulated via a subsequent FILTER.

The set of attributes A is now a set of variables V_A. In SPARQL a solution mapping is a partial function $\mu : V \to T$. Consequently, a variable might be undefined, i.e., have no associated mapping. We augment the scoring function to consider this case explicitly by treating undefined variables in a solution mapping as the worst possible value, i.e., $f(x) := \infty$ if x is undefined. Given these prerequisites, the semantics of the prefer graph pattern can now be defined.

Definition 1 (Prefer and Partition). *Let $Pref = (V_A, <_P)$ be a preference, G be a set of variables and Ω be a solution sequence of mappings $\mu : V \to T$. Then the preference selection operator is defined as:*

$$\sigma[Pref](\Omega) := [\mu \mid \mu \in \Omega \wedge \nexists \mu' \in \Omega : \mu[V_A] <_{Pref} \mu'[V_A]]$$

With PARTITION only the best solutions remain in each group:

$$\sigma[Pref\ PARTITION\ G](\Omega) := \left[\mu \mid \mu \in \Omega \wedge \nexists \mu' \in \Omega : \mu[G] = \mu'[G] \wedge \mu[V_A] <_{Pref} \mu'[V_A]\right]$$

Overall, the syntax is closely related to PreferenceSQL, except for a key alteration w.r.t. categorical preferences. Due to a grammar conflict on the keyword IN, a break with prior convention is required. The new keyword is ONE OF. Table 1 lists all available base constructors and Table 2 shows the corresponding grammar.

For convenience, PreferenceSPARQL supports the usage of SPARQL expressions in preferences. For instance, $?x+?y$ AROUND 0 is a valid preference which is equivalent to

$$\text{BIND } (?x+?y \text{ As } ?z) \text{ . } ?z \text{ AROUND } 0$$

Caution needs to be taken when using an expression as a goal value (e.g., low, up, d). The behaviour will be unpredictable, unless a constant value is used.

Table 1: Base Constructors in PreferenceSPARQL.

Preference Constructor	SPARQL Equivalent
Between$_d(V_x, [low, up])$	?x BETWEEN low, up, d
Around$_d(V_x, z)$	?x AROUND z, d
MoreThan$_d(V_x, low)$	?x MORE THAN low, d
LessThan$_d(V_x, up)$	?x LESS THAN up, d
Highest(V_x)	?x HIGHEST
Lowest(V_x)	?x HIGHEST
Layered$_m(V_x, [S_{1\to k}, others, S_{k+1\to m}])$	?x LAYERED (S_1, ..., others, ..., S_m)
PosPos(V_x, S_1, S_2)	?x ONE OF S_1 ELSE S_2
PosNeg(V_x, S_1, S_2)	?x ONE OF S_1 NONE OF S_2
Pos(V_x, S)	?x ONE OF S
Neg(V_x, S)	?x NONE OF S
$S = \{s_1, \ldots, s_n\}$	$(s_1, \ldots, s_n)$

Table 2: PreferenceSPARQL Grammar.

Prefer	:=	'PREFER' BracketedPrefer PartitionClause?
BracketedPrefer	:=	'(' ParetoPreference ')'
PartitionClause	:=	'PARTITION' '(' Var$^+$ ')'
ParetoPreference	:=	Prioritization ('AND' Prioritization)*
Prioritization	:=	PrefAtom ('PRIOR' 'TO' PrefAtom)*
PrefAtom	:=	BracketedPrefer | (Expression (DiscreteAtom | ContinuousAtom))
DiscreteAtom	:=	Layered | Pos | Neg
Layered	:=	"LAYERED" '(' (ListOfSets ',')? 'others' (',' ListOfSets)? ')'
Pos	:=	'ONE' 'OF' Set (('ELSE' Set) | ('NONE' 'OF' Set))?
Neg	:=	'NONE' 'OF' Set
ListOfSets	:=	Set (',' Set)*
Set	:=	'(' Expression (',' Expression)* ')'
ContinuousAtom	:=	Interval | Around | Highest | Lowest
Interval	:=	(('BETWEEN' Expression ',' Expression) | ('MORE' 'THAN' Expression) | ('LESS' 'THAN' Expression)) (',' Expression)?
Around	:=	'AROUND' Expression (',' Expression)?
Highest	:=	'HIGHEST'
Lowest	:=	'LOWEST'

4.2 Implementation and Query Execution

For our implementation and evaluation, we have chosen Apache Jena[2] and its SPARQL query engine ARQ (SPARQL 1.1. compliant). Apache Jena is a free and open-source Java framework for building Semantic Web and Linked Data applications.

ARQ parses a query and generates an Abstract Syntax Tree (AST). Next, the AST is compiled into SPARQL algebra as described by the SPARQL specification. The algebra generator uses Pareto to put multiple PREFER clauses inside a single basic graph pattern (BGP). ARQ then optimizes the algebra via high-level algebraic transformations. This includes a rewriting of the algebra into new, equivalent algebra forms and introducing specialized algebra operators. The algebra is expressed as a SPARQL S-Expression (SSE), a custom syntax for stating SPARQL algebra in a concise format. Subsequently, a query plan is computed. This query plan is then executed to get a solution sequence.

Execution of a *P* PREFER *Pref* PARTITION *G* graph pattern is realized by first partitioning the solution sequence of *P* according to *G*. For each partition, the requested preference algorithm is called separately. The computed solution sequences $\Omega_1, \ldots, \Omega_n$ are subsequently concatenated ($\Sigma_{i=1}^{n} \Omega_i$) in order to arrive at the final solution sequence.

[2]Apache Jena: https://jena.apache.org

5 EXPERIMENTS

In this section, we present comprehensive experiments on different preference algorithms for the evaluation of PreferenceSPARQL on RDF data. We adapted well-known algorithms (cp. [3]) from relational databases to the Semantic Web. For this we used BNL [2], LESS [6], SaLSa [1], and Query Rewriting. Please note that related work like [11, 13, 16] do not provide experiments or only consider BNL. Hence, we are the first which compare different algorithms for the evaluation of preferences in SPARQL.

The first three algorithms are improvements of the nested-loop algorithm, that compares trivially every tuple with each other. Query Rewriting transforms PreferenceSPARQL into plain SPARQL queries. The evaluation is constrained towards a quantitative approach, i.e., a qualitative evaluation does not take place. Furthermore, we use PreferenceSQL as a reference to better assess the practicability of our implementation. Note that there are more sophisticated algorithms than BNL and its variants. Most of them require a preprocessing step (e.g., indexing), which leads to an unfair comparison. Therefore, we decided to use well-established algorithms.

5.1 Data Set

We decided to generate synthetic data derived from real-world facts based on real estate purchases and sales. The general structure of the test suite is inspired by the well-known Berlin SPARQL Benchmark[3]. The data set is scalable and comes in two distinct representations. One uses RDF triple data that is intended for our PreferenceSPARQL client. The other one is for PreferenceSQL in the form of a relational data model. The data model consists of 9 coherent tables. One table, e.g., describes an agent who is responsible for a collection of sale offers. Other tables have information about the property, location, usage type of the land or internet availability. We came up with 20 queries that reflect the differences in preference usage as appropriate for each use case. Table 3 describes all queries and contains a rough estimation of the complexity.

Due to limited space, we selected four queries out of the 20 to illustrate our results. Those four queries and their SPARQL equivalent are shown in the Appendix. All queries can be downloaded from GitHub[4]. The test queries reflect different performance profiles. All queries are parameterized (@parameter@), in order to prevent caching.

We use Query 5 (cp. Table 3 and Figure 4) as an example that we explain in detail. An investor could use this query to look for properties that have a stable renter base, i.e., renters who pay on time. Equally important might be that the rental conditions compare favourably to market, i.e., the net rental return is higher than average. In order to retrieve all the necessary information, we need to join three tables in PreferenceSQL. The preferring clause consists

[3]Berlin SPARQL Benchmark: http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark
[4]Queries: http://www.preferenceSQL.com/Download/PreferenceSPARQL-Queries.zip

Table 3: Queries with dimensions and short descriptions. The dimensionality roughly indicates the complexity of the preference, e.g., '2/4' denotes a binary Pareto prioritized over a quaternary Pareto.

Query	Dimensions	Preference
01	2	Lots with the lowest price and enough space for the construction of a single-family house.
02	3	Properties with highest guide value, most residential units and most recently modernized.
03	2/4	Properties with certain target size and a certain number of residential units. Less importantly, the price, energy consumption and two other categorical characteristics.
04	1	Lots around a specific area.
05	2	Properties with an above-average net return and punctual rental payments.
06	2/2	Properties with a certain size and amount of residential units. Less importantly, price and the year of the last modernization.
07	3	Agricultural land with three specific numerical characteristics.
08	4	Properties with specific categorical characteristics regarding the construction.
09	14	Properties with eight specific categorical and six numerical characteristics regarding general facts.
10	4/2	Properties with four specific characteristics regarding the neighbourhood. Less importantly, are two specific numerical characteristics.
11	5/1	Agents with four specific sale characteristics before the lowest possible commission.
12	1/1/4/1	Municipalities with a range of different and equally important numerical characteristics.
13	2	Municipalities with a low photovoltaics adoption rate and high yearly returns per square meter.
14	1/7/1	Properties built before 1970. Less importantly, seven specific characteristics regarding the interior and, least importantly, the number of amenities.
15	1/3	Warehouses that exceed a certain capacity before three specific numerical preferences.
16	6	Properties with six characteristics that are similar to another property's characteristics.
17	2 partition 2	Lots with the highest internet upload and download rate for each municipality and internet type.
18	3/2	Properties with three certain electrical requirements. Less importantly, the construction year and the condition of the building.
19	1/1/1/1/1	Properties with five specific characteristics that have all different importance.
20	1 partition 1	Sale offers with a certain price compared to their market value grouped by the municipality.

of two base constructors (MORE THAN, LAYERED) and one complex constructor (AND) to combine them. Therefore, the dimension in Table 3 was set to 2.

```
select s.id, s.price_eur, (..)
from sale_offer s, property p, contract c
where p.id = s.property_id and c.id = p.contract_id
preferring c.net_rental_return more than @net@
        and c.payment_behavior
        layered(('Punctual'), ('Unknown'), others));
```

Figure 4: Query 5 in PreferenceSQL.

In Figure 5 we show the corresponding PreferenceSPARQL query. The query is more verbose, but the prefer clause is very similar to the clause in PreferenceSQL.

```
prefix: <http://example.com/real-estate/>
prefix xsd: <http://www.w3.org/2001/XMLSchema#>

select ?sale_offer ?price_eur (..)
where {
  ?sale_offer a :sale_offer ;
                :price_eur ?price_eur ;
                :commission ?commission ;
                :property ?property .
  ?property :contract ?contract .
  ?contract :net_rental_return ?net_rental_return ;
            :payment_behavior ?payment_behavior .
  prefer( ?net_rental_return more than @net@
        and ?payment_behavior
        layered (('Punctual'), ('Unknown'), others))
}
```

Figure 5: Query 5 in PreferenceSPARQL.

5.2 Experimental Setting

All experiments have been conducted on a standard PC (Intel i7-4770K 3.9 GHz CPU, 16 GB RAM, Windows 7 x64). The JVM has been assigned 10 GB RAM (-d64 -Xms10G-Xmx10G). The backend database for PreferenceSQL is PostgreSQL 9.4, which has been deployed to the same computer with factory defaults. The PreferenceSPARQL implementation is built upon Apache Jena 3.7.0 with default settings. The spill factor is set to 100 000 for externalized BNL.

We utilize TDB2 as our native triple store. Memory is bounded towards JVM-assigned RAM. Hence, performance estimates are optimistic. For a fair comparison to PreferenceSQL, TDB would have to be limited to the same amount of memory. A test driver dispatches preference queries to the intended recipient and collects benchmark metrics. Specifically, all metrics have been derived from execution time:

- *Query Execution Time (QET)* which denotes the time required to serve a query request (from dispatch till the reception of all solutions).
- *Aggregated Execution Time (AET)* which indicates the run time of all 20 individual queries by summation of their QET. Downtime spent by the test driver to prepare the next query is not measured.
- *Thread Execution Time (TET)* is the sum of all AETs of an algorithm for one query.

For statistical robustness, we use box-and-whisker diagrams. QETs and AETs are broken down according to lower quartile, median and upper quartile as indicated by the box. Individual points depict outliers.

5.3 Results

In this section we discuss the results of our experiments.

5.3.1 Query Execution Time by Algorithm (QET). **Figure 6** shows the query execution time of all algorithms. At the top is Query Rewriting, then LESS, SaLSa, the externalized version of BNL (BNL-E), the in-memory version of BNL (BNL-M) and finally PreferenceSQL.

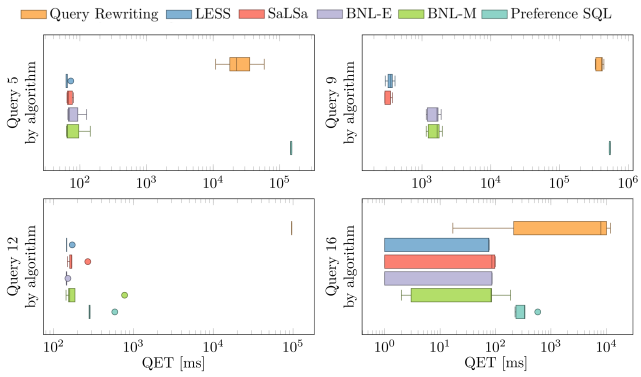


Figure 6: Algorithm comparison by single queries (100 agents, 5 iterations, single client)

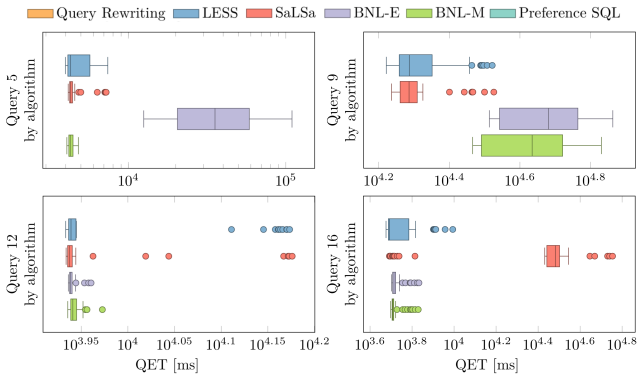


Figure 7: Algorithm comparison by single queries (5000 agents, 50 iterations, single client)

The data set refers to 100 agents that correspond to 297 834 RDF triples. Query 5 was already discussed above and has low complexity. The others have high complexity. Query 9 uses multiple Pareto preferences, Query 12 four Prioritizations and a nested quaternary Pareto preference and Query 16 multiple Pareto preferences with a nested lookup (cp. Table 3).

It is apparent that the performance of Query Rewriting is subpar. ARQ is not optimized for complex filter expressions. Hence, Query Rewriting should only be employed if no other algorithm can be used, e.g., due to a proprietary SPARQL engine. The performance of PreferenceSQL is mixed. On the one hand, it closely mirrors all native preference algorithms (Query 12, Query 16). On the other hand, some queries are exceedingly slow (Query 5, Query 9). The middleware architecture inherently slows down PreferenceSQL, especially when the whole data set needs to be retrieved.

Figure 7 focuses only on native preference algorithms. The data set was scaled up to 5 000 agents or 14 700 102 RDF triples. The performance of in-memory BNL is solid, mostly outperforming its externalized variant in simple queries (Query 5). In some circumstances, BNL is significantly slower than other preference algorithms, e.g. in Query 9, with multiple Pareto preferences. Choosing the correct window size is difficult, though, due to observed performance reductions with overly large window sizes.

Across all queries, LESS leads in terms of execution time, but the expected performance gain over BNL is absent. The spread in execution time for most queries is very apparent. SaLSa appears to suffer from the different choices made w.r.t. evaluation of the tuples on larger data sets (Query 16). Notably, LESS and SaLSa produce quite a lot of outliers. We presume this is due to the memory limitation of 10 GB, repeatedly triggering garbage collection.

5.3.2 Aggregated Execution Time by Algorithm (AET). **Figure 8** aggregates the QETs for all test queries. For 100 agents, the distribution of the AETs affirms our prior judgment that Query Rewriting is not suitable for larger workloads. The previously discussed non-performing queries skew the AET for PreferenceSQL. The AET for the well-performing query subset of PreferenceSQL is approximately equivalent to the native PreferenceSPARQL algorithms. For 5 000 agents, in-memory BNL outperforms its externalized cousin. SaLSa is roughly equivalent to in-memory BNL by median but gives rise to lots of outliers in both directions. Finally, LESS outperforms everyone else by the median, but just like SaLSa suffers from a large spread in execution times.

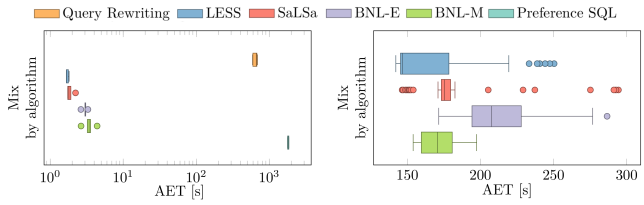


Figure 8: Algorithm comparison by aggregated execution time (Left: 100 agents, 5 iterations. Right: 5 000 agents, 50 iterations.)

5.3.3 Thread Execution Time by Algorithm (TET). The evaluated data set sizes range from 100 to 5 000 liaisons that are 97 834 to 14 700 102 triples, respectively. **Figure 9** shows the overall run time of the native PreferenceSPARQL algorithms across these data set sizes. From 100 to 1000 agents, the performance of LESS is excellent. SaLSa is closely behind LESS. Externalized BNL is the slowest method, but not far behind its in-memory counterpart. The gap between in-memory LESS and BNL rapidly closes as the size of the data set further increases. With 500 agents LESS only spends 58.63 % of BNL-M's run time. With 5 000 agents LESS already requires 97.19 %. Given the memory usage pattern observed during the evaluation, ARQ appears to run into the 10 GB memory limit. With a machine that has more memory, LESS should lead significantly in performance. An externalization for LESS and SaLSa should alleviate this problem in general.

6 SUMMARY AND CONCLUSION

In this paper, we introduced PreferenceSPARQL for querying the Semantic Web with preferences. We defined a grammar for using preferences with RDF data and built a benchmark derived from real-world facts in the domain of real estate purchases and sales. Our experiments show that Query Rewriting is no choice for evaluation,

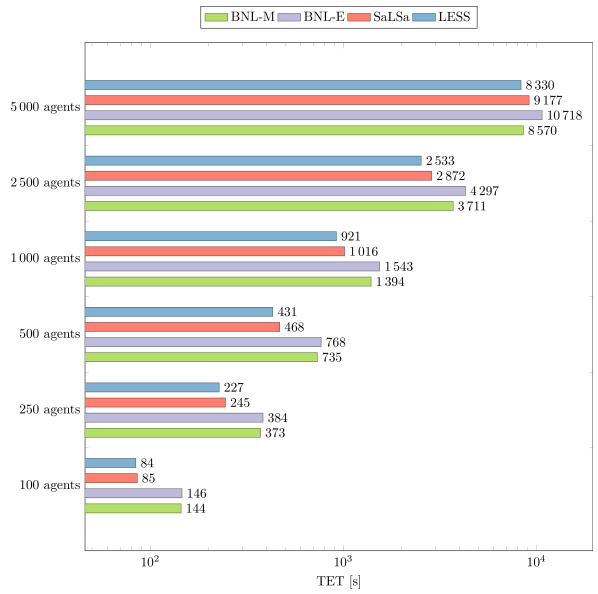


Figure 9: Algorithm comparison by overall run time across various data set sizes (50 iterations, single client)

because ARQ is not optimized to handle complex filter expressions. A native algorithm like LESS should be used.

As future work, we plan to develop more sophisticated evaluation strategies, and we want to consider qualitative aspects. Furthermore, the consideration of ontological knowledge or the formulation of preferences on the actual structure of the RDF graph might speed up preference evaluation significantly.

APPENDIX

Here we present the discussed test queries Query 5, 9, 12, and 16. All queries and their SPARQL representation can be downloaded at http://www.preferenceSQL.com/Download/PreferenceSPARQL-Queries.zip

Query 5

```
select s.id, s.price_eur, s.commission, c.net_rental_return, c.payment_behavior
from sale_offer s, property p, leasing_contract c
where p.id = s.property_id and c.id = p.leasing_contract_id
preferring c.net_rental_return more than @net@ and c.payment_behavior layered
(('Punctual'), ('Unknown'), others, ('Intermittent'), ('Overdue'));
```

Query 9

```
select s.id, s.price_total, lot.immissions, lot.shape, lot.neighborhood, i.type, i.download, i.upload, p.area_per_unit,
p.year_of_modernization, p.energy_consumption_ratio, p.heating, p.cooling,
p.condition, p.interior
from ex_sale_offer s, property p, lot, internet_availability i
where p.id = s.property_id and lot.id = s.lot_id
and lot.internet_availability_id = i.id
and lot.zoning in ('W_Residential', 'M_Mixed')
preferring s.price_total lowest
and lot.immissions in ('Location_Appropriate', 'Reduced')
and lot.shape in ('Level')
and lot.neighborhood in ('City_Center', 'Periphery')
and i.type in ('DSL', 'Fiber')
and i.download highest
and i.upload highest
and p.area_per_unit between @area_min@, @area_max@
and p.year_of_modernization highest
and p.energy_consumption_ratio lowest
and p.heating in ('Wood_Pellets', 'Long_distance_Heating')
and p.cooling in ('None')
and p.condition in ('Excellent', 'Good', 'Fair')
and p.interior in ('Superior', 'Normal');
```

Query 12

```
create or replace temporary view query12_rent_markets as
(
select loc.municipality, lot.zoning, count(*) as sale_count, avg(s.price_total)
as price_total_avg, avg(lot.guide_value) as lot_gv_avg, avg(p.guide_value) as
property_gv_avg, avg(p.unit_count) as unit_avg, avg(c.net_rental_return) as
net_return_avg
from ex_sale_offer s, lot, location loc, property p, leasing_contract c
where s.lot_id = lot.id and lot.location_id = loc.id and s.property_id = p.id
and p.leasing_contract_id = c.id
group by loc.municipality, lot.zoning
);

select m.*
from query12_rent_markets m
preferring m.sale_count more than 15
prior to
m.net_return_avg more than 0.03
prior to
(
m.sale_count highest
and m.lot_gv_avg highest
and m.property_gv_avg highest
and m.unit_avg highest
)
prior to
m.net_return_avg highest;
```

Query 16

```
select p.id, p.style, p.exterior_wall, p.construction, p.foundation, p.flooring,
p.condition
from property p
where p.id != @desired_id@
preferring p.style in (select style from property p where p.id = @desired_id@)
and p.exterior_wall in (select exterior_wall from property p where p.id =
  @desired_id@)
and p.construction in (select construction from property p where p.id = @desired_id@)
and p.foundation in (select foundation from property p where p.id = @desired_id@)
and p.flooring in (select flooring from property p where p.id = @desired_id@)
and p.condition in ('Excellent') else ('Good', 'Fair');
```

REFERENCES

[1] I. Bartolini, P. Ciaccia, and M. Patella. SaLSa: Computing the Skyline Without Scanning the Whole Sky. In *Proceedings of CIKM '06*, pages 405–414. ACM, 2006.

[2] S. Börzsönyi, D. Kossmann, and K. Stocker. The Skyline Operator. In *Proceedings of ICDE '01*, pages 421–430. IEEE, 2001.

[3] J. Chomicki, P. Ciaccia, and N. Meneghetti. Skyline Queries, Front and Back. *ACM SIGMOD Record*, 42(3):6–18, 2013.

[4] J. Chomicki, P. Godfrey, J. Gryz, and D. Liang. Skyline with Presorting. In *Proceedings of ICDE '03*, pages 717–816. IEEE, 2003.

[5] B. Glimm. Using SPARQL with RDFS and OWL Entailment. In A. Polleres et al., editors, *Reasoning Web. Semantic Technologies for the Web of Data*, pages 137–201. Springer Berlin Heidelberg, 2011.

[6] P. Godfrey, R. Shipley, and J. Gryz. Algorithms and Analyses for Maximal Vector Computation. *The VLDB Journal*, 16:5–28, 2007.

[7] M. Gueroussova, A. Polleres, and S. McIlraith. SPARQL with Qualitative and Quantitative Preferences. In *Proceedings of OrdRing '13*, pages 2–8. CEUR-WS.org, 2013.

[8] A. Hadjali, S. Kaci, and H. Prade. Database Preference Queries - A Possibilistic Logic Approach with Symbolic Priorities. In Hartmann and Kern-Isberner, editors, *Foundations of Information and Knowledge Systems*, volume 63, pages 291–310. Springer, 2008.

[9] W. Kießling. Foundations of Preferences in Database Systems. In *Procedings of VLDB '02*, pages 311–322. VLDB Endowment, 2002.

[10] W. Kießling, M. Endres, and F. Wenzel. The Preference SQL System - An Overview. *Bulletin of the Technical Commitee on Data Eng., IEEE*, 34(2):11–18, 2011.

[11] P. F. Patel-Schneider, A. A. Polleres, and D. Martin. Comparative Preferences in SPARQL. In C. Faron Zucker, C. Ghidini, A. Napoli, and Y. Toussaint, editors, *Knowledge Engineering and Knowledge Management*, pages 289–305, Cham, 2018. Springer International Publishing.

[12] O. Pivert, O. Slama, and V. Thion. SPARQL Extensions with Preferences: A Survey. In *Proceedings of SAC '16*, pages 1015–1020. ACM, 2016.

[13] W. Siberski, J. Pan, and U. Thaden. Querying the Semantic Web with Preferences. In Cruz et al., editors, *The Semantic Web - ISWC 2006*, pages 612–624. Springer Berlin Heidelberg, 2006.

[14] K. Stefanidis, G. Koutrika, and E. Pitoura. A Survey on Representation, Composition and Application of Preferences in Database Systems. In *In ACM Transactions on Database Systems (TODS)*, number 36. ACM, 2011.

[15] H. Stuckenschmidt, F. Harmelen, W. Siberski, and S. Staab. Peer-to-Peer and Semantic Web. In Staab and Stuckenschmidt, editors, *Semantic Web and Peer-to-Peer: Decentralized Management and Exchange of Knowledge and Information*, pages 1–17. Springer Berlin Heidelberg, 2006.

[16] A. Troumpoukis, S. Konstantopoulos, and A. Charalambidis. An Extension of SPARQL for Expressing Qualitative Preferences. *CoRR*, 2017.

[17] M. Workman. Introduction. In *Semantic Web: Implications for Technologies and Business Practices*. Springer International, 2016.

[18] L. Yu. The Building Block for the Semantic Web: RDF. In *A Developer's Guide to the Semantic Web*, pages 23–95. Springer Berlin Heidelberg, 2014.

Pandemic and Big Tech

Bipin C. Desai*
BipinC.Desai@concordia.ca
Concordia University
Montreal, Canada

ABSTRACT

Having been an observer and user of computing devices from slide rules, analog computers, early monstrous digital machines, to sleek, hand held digital ones: seeing the shift of the computing and data 'ownership' paradigms over the last six decades one wonders at the enormous size, power and market capitalization of a fistful of companies that have existed for only a couple of decades. Now the world is groaning under the corona virus pandemic mismanaged by most governments, health officers and organizations. Are these not perfect examples, ad- infinitum of the Peter principle? At the same time big tech is benefiting from the pandemic and preparing to take a central role to harvest more data, to be mined in the future for more revenue streams. This paper looks at the recent push by big tech to push its agenda to reach into all aspects of human life. The current opportunity presented by the Covid-19 pandemic and the fear of future pandemics is being seized to lay the ground work, at the public's expense and their privacy.

CCS CONCEPTS

• **General and reference**; • **Software and its engineering**; • **Social and professional topics**; • **Applied computing**; • **Security and privacy**;

KEYWORDS

Privacy, security, smartphone, contact tracking, big tech, surveillance

ACM Reference Format:
Bipin C. Desai. 2020. Pandemic and Big Tech. In *Proceedings of 24th InternationalDatabase Engineering & Applications Symposium (IDEAS 2020).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410585

1 INTRODUCTION

Like mobile phone technology, the internet and the web, which have been exploited by new players since the original existing players in place were restricted by legislation. For example, the national postal service should have been called in to provide email service to supplement the other postal service. The lack of political will and the resistance to providing funds to the postal service to build up the expertise and infra structure meant that this did not occur. Private capital was able to support the emergence of big techs since there were great fortunes to be reaped. The other problem was of course the lack of imagination of complacent management of the existing original corporations to provide the additional services and politicians to prevent, nay promote, monopolies in these new arenas. This was what prompted capital to be made available to the emerging robber barons of the late 20th century. These corporation headed by buccaneers started putting down their own rules and bought politicians to stay out. Their big purses allowed them to bend most politicians and anyone with independent thought and ideas was put down by the anti-populist forces[24].

1.1 Pandemics

The latest epidemic to hit humankind is the severe acute respiratory syndrome corona-virus 2 (SARS-CoV-2) and it spreads from person-to-person through close contact[2]. When an epidemic from one country crosses a country's boundary, it is called pandemic. Throughout history [33], a number of pandemics have been recorded. Some from the ones recorded and observed were: the 14th century Black Death which is said to have stopped the Vikings conquest of the new world. This was followed by re-discovery of the new world and spread of smallpox, measles and bubonic plague by the Spaniards and other Europeans looking for treasures in the new world. These epidemics were foreign to the existing people and wiped out 90% of the indigenous population of North and South Americas. Whatever remained were further decimated by the settlers in the Americas.

In the 20th century, it is recognized that human-kind had suffered three influenza pandemics. These were the 1918-1919 Spanish Flu[22]; it started in Europe and USA and spread around the world; it lasted till 1919 when community immunity was said to be developed. The 1918 flu is said to have caused 25-50 million human casualties. This was followed by the 1957 flu pandemic, also called Asian flu pandemic of 1957 or Asian flu of 1957. This outbreak of influenza was first identified in February 1957 in East Asia and subsequently spread to countries worldwide. The 1957 flu pandemic was the second major influenza pandemic to occur in the 20th century; it followed the influenza pandemic of 1918–19 and preceded the 1968 flu pandemic. The 1957 flu outbreak caused an estimated one million to two million deaths worldwide and is generally considered to have been the least severe of the three influenza pandemics of the 20th century. The third pandemic of the 20th century was the 1968 flu also called the Hong Kong flu. According to [41], its introduction on the west cost of USA following it caused a high number of infections and mortality with the global mortality estimated between one to two million.

As was found during the Spanish flu, the biggest factor in its control was social distancing as illustrated in a recent article[74].

*Corresponding Author

IDEAS 2020, August 12–14, 2020, Seoul,Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410585

This simple measure, before the era of computers, Internet, web, OSNs, AI, Big Data or big techs just required the acceptance of this precaution by the people. This was the approach used during the current pandemic in Sweden. The result of their approach as compared to those done in other parts of the world will have to be determined by a more careful study when the data is in. Preliminary studies which added one factor, viz. masks, has come under fire as not being scientific and objective. However, one cannot possibly do a controlled experiment – one can only report from observation!

The use of mobile phone application to track people may help in locating the closeness of one device to another suspect one and may or may not be a violation of the social distancing rule. However, like all devices they could be manipulated by the users and is after the fact. Just as the body camera mandated by some jurisdictions and the presence of watching eyes has not prevented the militarized police forces from acting in the way they have been doing in many parts of the world. The most notable examples are being reported in the press.

1.2 Origin of the novel Corona-virus

The origin of Covid-19 has gone through a number of claims, counter claims, speculation, and stories about deliberate leaks and accidents. However, the scientific communities have concluded its origin like many other infectious diseases to be cross species [10]. One of the factors that has to be considered is that over the last 50 years, the earth's human population has doubled without an appreciable increase in the living space. This increases the interaction and the increased mobility gives rise to transmission of infectious diseases. It is reported that up to five million people travelled out of the epicentre of Covid-19 before March 2020 to all parts of the world. Anyone who would have come in contact with these travellers also became secondary and tertiary carriers. From various news sources, Nov. 2019 is the likely first case of Covid-19 and ireported to have occurred in Wuhan, China. More cases were reported in Dec. 2019 and attributed to an unknown virus. When it was observed and reported that the virus was being transmitted by human contact, the physician reporting it was charged with spreading rumours and forced to 'confess'. He later died from the virus[94].

Even though some health authorities were aware of the virus, the lock-down began outside China only in mid-March. However it took even longer in some parts of the world than others. The number of people infected and developing the disease has increased at an alarming rate. By the end of June there seemed to be a decline in the daily new cases in many parts of the world. Some areas, where the lock-down was lifted too early has gone through a resurgence.

2 HOW DID BIG TECH BECOME BIG?

A number of massive technology-based companies have evolved; some of which are not the age of majority! The conditions for their development stem from two phenomena: the un-bundling of computing software from computer hardware, and the emergence of an application of the internet. One of the current big tech firms got its start when IBM went in search of an operating system for its personal computer. An unknown little company was awarded the contract and this company went and bought one for a paltry sum, and re branded it[90]. This company is now one of the frightful five[49]. As indicated by this event and the one discussed below, these companies have no software and if there is a fortune to be made putting together something to be improved in the next version and the next version (we know we have lot of work to do!). In the meantime one has to put up with buggy unsatisfactory products.

2.1 Software and Hardware

The marketing of computing systems in the early days included the bundling of basic software support. This included the operating system, the compilers and libraries as well as training manuals. An organization would either buy the bundle or lease it and develop the specific software applications for its own use in-house. Computer Science evolved to train people who would develop this application software. The competitors to IBM, the most successful manufacturer of bundled systems were forced by early software only houses, using the courts and USA's anti-trust laws to un-bundle software from the hardware. The anti-trust case was based on the rationale that people who wanted software should not have to buy the hardware as well. This anti-trust case was finally dropped but it gave rise to a number of software houses. This and the ideas of one size fits all concept led to the establishment of software houses which produced sets of generic software that could be used for many businesses and replaced the in-house systems. This un-bundling and the introduction of the PC by IBM which was an un-bundled hardware system gave rise to the birth of one of the what today is big tech five sometimes called the fearsome five[49].

In theory. in-house software could be abandoned and replaced by software from such companies at a tremendous cost savings. The employees could be let go and only a small number would be required to tweak the software which would be maintained by the software house. The software house would have a stream of revenue for providing service and update: it would be a win-win situation. From the experience reported in a report by the Canadian Senate and the Wikipedia page, one can see the fiasco caused by the system "Phoenix" bought by the Harper government to save money. After five years of continued complaints about underpayments, over-payments, and non-payments, due to a software system that was supposed to save $70 million a year, to fix Phoenix's problems it will cost Canadian taxpayers up to $2.2 billion by 2023 according to a Senate report

2.2 Internet, Web, Open Internet and Net Neutrality

The interconnection of computers through an internet has been around since the mid-1960s. It took a few years for email to become popular. Applications such as file sharing among academicians gave rise to applications such as Archie, Veronica etc. The audiences for these applications were generally limited to the academic and technical community. The introduction of the web, just yet another internet application, however had the potential of expanding the internet participation to billions of users gave rise to at least three of the fearsome five at the close of the twentieth century. The domination of these corporations is due to the practice of acquiring any potential rivals, and extending their reach in all areas of the network including cell phones. Two of these fearsome five are in the mobile market and have a lions share of the market through

the operating system used on the cell phone and/or marketing their own brand with limited lifetime and built in obsolescence. They have replaced the early pioneers of the smartphone market by giving away the software or having complete control over all aspects of the mobile phone from hardware, software and the store which distributes applications for their brands.

Net neutrality is the term used to signify that all internet traffic must be treated equally by all providers of the internet regardless of their content and origin. The providers include the internet service providers, the telecommunication companies and the equipment on them. Open internet is the term used to signify that all resources and operations on them must be accessible to all users of the internet; the users could be individuals. corporations and organizations both governmental and non-governmental.

Neither of these principals are 100 percent enforced everywhere. There is restriction in some countries and there are always other impediments such as pay walls and private sites etc. What this principle has fostered is the big-techs who provide so called free service to all users at a very high hidden cost: namely their personal data which is used in manipulating them either by products or services that are being promoted by their advertisers. Hence the personal data, given up by billions of unwitting users have been hijacked and used, shared, sold by big-techs who hide behind labels such as not being evil or connecting people. This is very much like the fact that poor countries contributed viral information for epidemics to big pharmaceuticals and the result of these viral analyses led to medicine that was not affordable to the countries that provided the data. Another example of personal data given for the imaginary peace of mind is the personal DNA reports from millions of curious people which was later sold to bio/pharma industries.

These free services have enabled some of these corporations to be powerful earning large sums of money for each of their users who are hooked to them.

One would expect that since they are controlling the 'application store' they wold have some diligence in ensuring the quality of the software they make available and take a percent of the sale price. Recent articles in the press have shown that some of this software, as usual have bugs and security loopholes which could be hacked by spyware makers. One of these is attributed to a spyware firm in Israel which has targeted activists[56].

The USA's antitrust law was instrumental in the break-up of Bell under the antitrust legislation; however this action took decades before the break up was actually done. This break up was to prevent vertical monopoly. The location data of millions of cell phone users are sold to third parties and they in turn, turn around and sell it to anyone including the surveillance states machinery [46].

Every time one of the big techs is caught with a security or privacy violation, hatred leading to silencing critics and leading to their convictions[57] and even genocide as in the case of Rohingas [37, 40, 52] attributable to their platforms they utter some version of the mantra "we know we have more work to do "but give no indication of how they would re-dress the injury nor address what, when and how! The issue is that some of these platforms are driven by holding onto the the user's attention and ignoring their own research findings that point out that emotive and divisive contents are the ones that will keep the eyeballs glued to their site! This is what earn them the most revenue.

The truth of the matter is that these platform are too big and have colonized the internet. The big-tech business model to get as many people as possible to spend as many hours as possible on its site or their device so that they can sell those people's attention to advertisers. The myth that the internet is free sis a farce. Each society, each city, each community must have their own contents under their own jurisdiction and control of accountable elected officers.

Some of the for-profit big-techs that run social media, make a claim that they support social justice; however, their product and their marketing models do not reflect this lip-service[35]. They claim that they spend billions of dollars for work on AI to address these problems, but their model uses the research which shows that divisive contents attract and keep the audience. Also, how much of these billions is to support tools to weed out objectionable material including hate speech, pedophilia and false claims[65]. The USA's administration is headed by one who is known for promoting "divide and conquer" practised by invaders over centuries. By inventing a tag such as "newsworthy" for any contents that violate accepted decorum but coming from some political figures is allowed because of its news value. The label does not focus on the inaccuracies or falsehood nor whether it is hateful[11].These platforms are addictive and targeted at people of all ages, very much like the gambling platforms described in[42] targeted at young teens..

The internet age led to the introduction of three of the fearsome five big digital technology companies all with control in the USA. It is true that there are similar giants in China, to date, their influence on the world is not as penetrating as the fearsome five. In spite of the language barrier, some of these five are trying to break into the Chinese market.

2.3 Track record of Software System and Big Tech

2.3.1 Software Vulnerabilities. Designing and building hardware and software are challenging tasks and require foresight, careful analysis, testing and examining all compromises. In the engineering design of physical systems, the safety factor denotes how much stronger the system is compared to the worst case load over its lifetime. Physical systems such as bridges and building could not be tested and hence detailed models and their analysis is done to ensure adequate factors of safety. No such measure exists in software design and implementation and installation.

It has been shown over and over again that both hardware and software could have issues which could lead to vulnerabilities. While hardware wears out and has a life span, software does not wear out. For software the vulnerability is due to issues in software which could be one or more of: flaws in design, flaws in implementation of the design, lack or errors in adequate security checks. Makers of software, to continue with a revenue stream, issue new versions of software and entice existing users with new features while ending the support of bug and vulnerability fixes beyond what they call the end-of-life. This is of course an artificial invention and it has led to one of the worst cyber-attacks in 2017[87, 93]. Vulnerabilities in software have lead to attacks of the computing system using worms and viruses. Many of these have been reported in the press and included WannaCry and ILoveYou

which targeted older versions of Windows and the then current versions which were not updated. Since updates of Windows, require shutting down all applications and rebooting, which is not possible especially for applications that need to be on-line all the time. Contrast this with open source and free Linus systems where most updates can be done on the fly. Cost of migrating to newer versions is another factor for many cash strapped organizations.

2.3.2 Lack of Social Commitment. The aim of the big tech is to dominate the market, claim user information as their own and exploit these data for extending their reach. A case in point is the contract to develop a track of land in Toronto harbour front. The RFP was dated March 2017 with a deadline of just one month. The contract was awarded in October 2017 to Sidewalk Labs an arm of Alphabet, the parent of Google which owes it birth to a search engine that was not even the first one since the concept was already developed and came out of one of the workshops during the first web meetings[4, 5].

In 2015 Google set out to improve cities and make them smart by launching Sidewalk Labs. One of the first contracts that this newly minted spin-off landed was the of Toronto Waterfront Quayside project in 2017. As expected the master plan, released in 2019 did what it was promised: it showed the vision of a city block as a product built using "internet up" paradigm, not people up. There had been a great deal of opposition to this project. As with many new projects involving software and computing system the project was not well thought out from the human and their social needs point-of-view. To the best of the authors knowledge, no software is written or building put up. From the track record of most software written to date one would expect it to be full of omissions, commissions and bugs. One should also expect rushed products which are many-a-times sloppy as evidenced time and again by recall, updates and re-writes, which obsolete old versions and require funds to move to new versions. Bugs security loopholes, built in or introduced by ignorance, remain in the old versions. The hazard of this was illustrated by the WannaCry attack. As pointed out in a Forbes article, there was not a single mention of citizens in the early report from Sidewalk Labs.

The final downside of using private corporations is that they will bailout if they feel that they are not making sufficient revenue out of a project. A case is point is the development project handed out by Toronto to Sidewalk Labs to develop a tract of land of 12 acres. This signing was done amid great fanfare by Justin Trudeau, and Eric Schmidt for a community built "from the internet up". The company expanded their project to develop 300 acres of prime waterfront property and wanted a share of the property tax. Also, in this instance, the Canadian Civil Liberty Union has sued the three levels of government[13]. The project is considered a private enterprise's attempt to colonize and use surveillance capitalism ignoring privacy and social issues. For example in the master plan released by Sidewalk Lab, there is no mention of citizens: and as reported in [79] '"innovation," "data" and "digital" are mentioned repeatedly, but other terms such as "social","culture", and "human" are relatively scarce. Self-driving cars are mentioned more than social infrastructure. Some words that one might expect to see used heavily in an urban planning report, such as "citizen," are barely mentioned at all'. This project was abandoned, without ceremony[15, 21, 38]!

Many of the ideas to build something from the internet up is questionable since internet is only a tool not a be-all - end-all. Simple fixes such as adjusting the timing of traffic lights on demand and prioritizing pedestrian crossing modes would not only be more rewarding than a 'smart' city which would be a dumb thing to do in spite of the gushing enthusiasm of marketers and politicians.

The phrase smart city in effect means the use of data collected by sensors and other devices including smart phones (another smart!), algorithms and technology (smart). However many of the features of a so called sustainable smart city, such as modular housing and wooden building high rises, are already in place. The large scale collection of data is filled with perils and requires public debate[85]. Another problem with the smart city model put forward by big-tech is the that these corporations are not satisfied with influencing politicians providing software and services to replace the ones in place in house, they want to be the governors without accountability and representation in a private for profit corporation. Unlike the old conquerors, they are coming not with weapons to plunder, they are taking over silently with the blessing of the politicians. Also, it is clearly felt by some advisers to the Waterfront project that the justification of why a digital approach is chosen over a non-digital one was not made in the smart city proposal.

3 PANDEMIC: BIG TECH STEPS IN

We know how big-tech and the fearsome five has been ignoring all going ahead and breaking things in their pursuit for getting the product out, to monetize user data; may this be from their emails, their browsing or their transactions. Even the banks are getting into this surveillance capitalism by mining the banking transactions. The mantra used is to provide service to the users by sharing their data with third parties none of which are named. The other justification used is to connect people and not be evil, the judge of this being themselves. Some of the heads of the tech giants use all opportunities[71]. As reported by Kline and others, to attract more public funds and used the pandemic to rehash their presentation[23, 44, 45]. These heads or ex-heads have financial resources and connections to set up organizations and/or participate in organizations to do so; one only has to look at the makeup of many of these commissions to discover the usual suspects from big techs and other interested corporations [54]. The message is for a partnership between the state and private industry to catch up with other nations; this in-spite of the billions these heads have amassed. The reward of any such partnership would be retained by the private industry who would try to escape taxes imposed on other working classes[1].

The danger of allowing big tech to enter into the tracking game for this and any future pandemic is well illustrated by the case cited in the Guardian of a company which had been fined recently for breaching privacy of user data and allowed to track the Corona-virus cases. It has already had to own up to not observing the data protection rules on its presently awarded contract for Covid-19 test-and-track[80].

[1]It is worth noting a comment by one of the readers (Tamza) of [71]: "I cant even bring myself to read such pieces from people who talk of 'no govt involvement' until they want more money. Use your outrageous [and tax evaded] earnings to prop yourselves up".

3.1 Contact tracing

Contact tracing is the tracing of the whereabouts of a known infected person and thus be able to warn these persons of the danger of potential infection[18]. Contact tracing generally involves first identifying, through a primary health provider, a person who has a communicable disease. This persons is contacted by the public health authority, to determine the person's contacts, movement and close relations. These other persons are also contacted. One of the more successful human centred contact tracing has been done in BC, Canada [19]. The persons and the close contacts are isolated. Other media, such as news stories, could be used to warn people if the contacts of the infected person could not be easily identified. This occurs if the infected person has taken public transportation and been in public places.

With mobile technology contact tracing has moved to that media and the makers of the both dominant operating systems for smart phones have released a contact tracing application interface (API). These API were released in April 2020 and are to be enhanced later in 2020. One wonders if these enhancement could have included omissions, commissions and of course bug fixes! In the meantime various protocols for privacy preservation are being debated. Among those being debated in EU includes: Pan-European Privacy-Preserving Proximity Tracing (PEPP-PT)[60]
which adheres to the EU privacy and data protection regime; and Decentralized Privacy-Preserving Proximity Tracing (DP-3T), a decentralized protocol[20] where each client's cell phone is assigned a semi-random identification named Ephemeral IDs (EID). When one client encounters another, they exchange their EIDs. The EIDs are not stored in any central location. However, when any client has tested positive, the client's EID is sent to a central site which is broadcast by it to all clients. Any client who was in contact with the infected client is thus made aware of the contact and hence could take corrective measures. The API s for Android and IOS is reported to use this latter protocol[20]. A number of organizations, including Amnesty International has issued a check list of any tracking application and this includes limited and justifiable tracking, preserving privacy, use restriction including use by third parties which have to be well defined with a sunset clause[3].

The smartphone market operating system is now controlled by just two of the frightful five; the pioneers of smartphone having been eliminated! The phone is meant to be smart so that people using them need not be! The key to this is having all data in the phone accessible to the operating system and applications in the phone. One of the features of these phones, to make it more attractive to a wider range of consumers, is that the company that controls the operating system of the phone allows others to develop applications for them and any application can access users data including contacts, pictures, passwords etc. The 'testing' and distribution of these applications for these phones, which makes them so convenient and hence popular, is controlled by these same companies. People get the false confidence in them since these applications are, in theory 'certified' by these two behemoths.

Smart-phones are currently the most ubiquitous "Internet of Things(IOT)" device. However, as it was pointed out in[6], the software for IOTs is rushed and full of errors bugs and back doors. A case in point is WhatsApp a very popular communication application available for both the smartphone OSs. Applications such as this should have been developed by the traditional telecomms who have some legislature control in most countries. However, the telecomms are too complacent and seem to lack imagination. The Israeli based company NSO had developed a system called Pegasus which was used vulnerability in WhatsApp to put a tracker in the smartphone so hacked and thence track the communications of the user of the hacked smartphone. According o [16] once Pegasus is installed it can contact the operator of the breach who can send commands and receive the complete set of data including all passwords, contacts, reminders, text and voice calls. In addition the operator could turn on the phone's camera and microphone, use its GPS to track the target. The targets of these hacking were activists and journalists[1]. NSO now has a software for Covid-19 tracking. called Fleming.

Also in April 2020, Google and Apple announced that they would suspend their rivalry to work with nations of the world to create a new alert system. They would reconfigure their mobile operating systems, incompatible by design, to notify users if they have stepped within the radius of a device held by a Covid-19 patient. An application of this could be the following: the history of contact of a person's phone could be used to help demonstrate the person's fitness to return to the office or board a flight.

StopCovid is a 'centralized' corona-virus tracing application launched in France. In a centralized version the data is transferred to an external system for storage. The data could include: everything on the cell phone such as contact list and Geo-location information, which gives where and when the cell phone owner has been. This naturally is a cause for concern in a free society and could lead to mass surveillance. The effectiveness of the application depends on when the user, who has downloaded and installed the application, develops the symptoms of Covid-19. Gets tested in time and would report it voluntarily to the application. The centralized system would then alert all other user who came in contact with this person if they had also installed and activated the application. One wonders the trail of this data, how is it safeguarded ans who would be able to access and use it!

Corona-Warn-App is the one launched in Germany- it is a decentralized approach and has additional security measures. It is said to be more successful and just like an application launched in India has been downloaded and installed by millions. The success of such an application, in a free society, is dependent on everyone in a public place having a cell phone which is activated with the tracing software. Even though it is imposed in some countries, it is an option in a free society. It has been reported that after about three weeks, the Covid-19 tracking application called StopCovid, launched in early June 2020 in France has alerted about a dozen people and five times that number have tested positive on the application. Of just over a million downloads and installations of the application on their mobile device, about half have removed it. Also, the symptoms of Covid, which are similar to symptoms of a common cold and flu, at least in the early stages, have to be correctly called by the user. This is also an issue.

3.2 Mobile API - Typical Requirements

Letting yet another application access more data on mobile phones than required could lead to security risks and expose personal information to unknown third and parties. By design, almost all mobile applications transmit and receive data between phones and remote servers. It has never been more crucial to understand the risks involved in giving mobile applications indiscriminate access to your personal data and sensors which include the GPS, camera and microphone.. A lot of this unnecessary access requirement also has to do with how these applications are built and monetized. To make money out of applications, companies often integrate third-party libraries that allow these external entities to push publicity and other content to the mobile device. Attackers can leverage poorly written code or third-party libraries to gain access to a user's phone sensors and data which include all photos, contacts and messages, Some of them use these contacts to send out emails and or text messages inviting these contacts, in the name of the device owner, to join this or that organization or group.

The fact that the cell phone owner is also the owner of these data which should be used only if the owner approves it, for each use. If stored, for backup it should not be disclosed to third parties and should be deleted when requested. There must be a legal requirement for the length. It must not be shared with a 'third party' as some banks want to do with their customers online transactions and other data. It is not always easy to know what data a specific application will access, or how it will be used, shared and sold. Most people do not consider the following: what is known about the application; who created it and what it does; what data it downloads and how does it abuse/use it!

A typical user, when using her desktop or laptop, uses the applications she wants. The applications runs as the user and has access to what it needs from location she specifies. Applications on cell phone are running with the personality of the maker of the application or the OS and never ask for permission to any piece of data or functionality. None of these should be sent out to servers anywhere. Only the information that is chosen and sent by the user should be transmitted to the destination chosen by the user. Any other access is not justified regardless of the platitudes of the software creators.

Regardless of the use of a decentralized contact access protocol, the application developer could easily connecting an EID to the cell phone number and thence the person. In addition to the application developer, the supplier of the phone and its operating system and the ubiquitous third party can have access to this confidential information. The dangers of third party has been well illustrated by the exploit of the defunct Cambridge Anlytica [30]. It should be noted that not everyone has a cell phone Currently in mid-2020, the number of people having smart phone is less than 46% and of any kind of cell phone is less than 62%; even in the developed countries less than 80% of the population has smart phones. Most applications on cell phones using different OS are not inter-operable. Hence the contact tracking solution is of limited use.

The other big issue is the foundation of contact tracing. Any success of it requires that the person a is not asymptomatic and s/he is tested as soon as symptoms develop. As it is the testing record in even most developed countries have been dismal. So even an efficient contact tracing via cell phones or manually by human calls is of little use: without widespread and rapid testing, it won't make much of a difference.

3.3 Low Tech Solution

Social distancing as was the solution in the Spanish flu-pandemic with better public education which does not depend on privacy invasion and handing over personal information to data gangsters. In spite of the fact that there was a lock-down in Wuhan, the epicentre of the Covid-19 virus, millions of people were allowed to travel out of the city[31]: this was not social distancing for the people at the destination of these travellers. The health authorities in many countries including the World Health Organization discouraged the use of masks. There was much publicity about how to wash hands for 20 seconds but no mention about the face or masks. Could the virus have landed on the face and even cleaned hands could transfer them? The author sees no mention of washing the face after washing the hands anywhere! If one is exposed to a cloud of virus during ones passage in a public place, even with masks, a healthy number of specimens may remain on the face. Human habit of touching the face could transfer these specimens to the inside. Does it not make sense, to wash the face as well?

As mentioned earlier, some EU countries have adopted the decentralized approach for contact tracing. However, it has been reported that the application on the mobile phone powered by Android to work, the location setting must be turned on. This enables this operating system's maker to know the mobile phone's and hence the owner's location. The record of data collected by one of the biggest pirates in the history of mankind of personal information is well recorded. Many users would not have trusted in this application. Any misuse of the location information so gleaned would be a mockery of the privacy promises that governments made to the public[55]. Another mobile application with a security flaw is the one in S. Korea: it is reported that this flaw has been fixed[70].

Until summer 2020, most health organizations discouraged masks and some continue to do so! Some rationale for this was put forward, by charitable commentators, such as there would be a run on masks and there would not be any for health workers, people would touch the face to adjust the mask and thus would self-infect, etc. etc. However, later, these same health authorities, who should have known better, changed their mantra. One can only imagine the number of unnecessary fatalities which could have been avoided if this blunder was not made. These are people who are supposed to be competent. As it is reported in a controversial article, the wearing of masks made a difference in lowering the increase in the daily new cases[14, 84, 95]. Other research has shown that wearing a mask controls the spread of the virus[28].

The hi-tech contact tracing solution depends on the person having a smartphone of some kind and it has the contact tracing application loaded and operational; it also depends on easy access to rapid testing and reporting of suspect cases. Access to testing, even in the developed countries, is shameful. Furthermore, even in the developed countries less than 85% of the population has smartphone and many of them are unable to use all its functionalities or be able or willing to download and install the contact tracing

application. In the developing world some 60% of the population do not have a smart phone[92]!

Putting the contact tracing application in the operating system in fact gives the maker of the cell phone operating system the unfair advantage of access to yet another set of data. The big tech has been mining users data for making a fortune and the users are addicted to the ease with these devices and products. In reality, none of the users data should be accessible to anyone: not to the maker of the mobile operating system or any of its applications. This is going to take a long time before some legislation is actually going to force these giants to be reined in.

However, as illustrated in two countries, Japan and Taiwan, a very simple strategy of everyone wearing a mask was used. This strategy does not depend on any of these, yet to be fully developed, technical solutions. Evidently the technical solution requires loading of billions of smartphones with the correct application and trusting private health data to the supplier of the software. Their track record has not been great; they have been mining user data from day one. A suggestion that has been made is the following: instead of having the user download and install the application the two makers of the smartphone operating system would make tracing application part of their updates and not give the owner of the device an option! This may not be an acceptable option. Wearing of masks by everyone could have contained the spread.

Low tech contact tracing, as in the one with mobile phones, involves timely testing of suspect cases and using humans to contact these cases and to determine the persons contacted by the cases before the onset of the symptoms then contacting these exposed persons[32]. It is also important that the suspect cases be in isolation. For example in some countries, there is mandatory supervised isolation of all new arrivels and of entire populations in those areas where the daily cases surge up[8].

Human contact tracing, as used in some of the provinces in Canada, adds the advantage of following a stricken person, guiding her/him to the correct resources specially when the person is alone and in quarantine: for example students away from home and seniors. This strategy calls for the training of more personnel to handle increased caseloads which include the most vulnerable. This also requires that a public health maintenance system should be in place as it has been in Canada or many parts of the EU.

Since it is impossible to conduct a double blind tests on random samples in a pandemic, the only other avenue open is observations of countries where masks are accepted and others where they are considered objectionable. A study published in The Lancet[95] provides a more balanced conclusion and includes face masks in the control armour along with social distancing and washing of hands[14].

The simple solution adopted by many East Asian countries was the donning of masks. The reduction of the probability of infection goes down when both parties are wearing a simple face mask. It is ironic that the advice given in the early days of the lock-down by the chief medical office in Canada was not to use face-masks. This was extremely imprudent and irresponsible advice and one cannot know how many people would not have died if a proper advisory for face mask use was given early on. Canada and it's chief medical officers were not alone in this fiasco. As reported in NY Times[66], the advice on masks from international experts was mixed, if not outright skeptical. For example, the surgeon general in USA sent out a stop buying masks tweet. The Center for Disease Control and Prevention of the USA's initial position was that if one is not sick, it is not necessary to wear a mask.

Another observation was reported in [81]. Taiwan with a direct flight from Wuhan was able to avoid a Corona virus-19 epidemic. The simple solution they adopted was wearing of masks. The total cases reported until the end of May was less than 500 with only 7 fatalities. This seemed to have been achieved without a lock-down. The mobile application created in Taiwan was to locate the nearest supply of masks! The production of masks went from two to 20 million. With the potential source and victim wearing masks and keeping social distance the risk of transmission is much lower; no one would be able to give a figure for this probability; any value could be challenged due to lack of scientific evidence and control testing.

At the start, Japan had inadequate social distancing with packed public transit. However, Japan, along with most oriental counties has an acceptance of wearing a face mask to contain the spread of communicable diseases including colds and flu and inhaling of dust, pollen or pollutant. The Jain community has a tradition of wearing a mask to prevent breathing in and killing of microorganism and small insects. However, it also acts as a deterrent form transmission of contagion through breath. As in other oriental countries the Japanese had used masks during the SARS and MERS outbreaks. During the current corona virus pandemic, researchers established a close correlation between high levels of mask-usage and slow levels of spread of the virus[66].

Masks also played a role in controlling the spread of the disease in Japan: its record to end of June 2020 was less than 20,000 cases and 1000 fatalities. Japan has a population of over 125 million whereas Taiwan's population is over 23 million. Compare this with the UK with a population of over 66 million with over 280,000 cases and close to 45,000 fatalities to the end of June. 2020.

4 CONCLUSIONS PREVENT DISASTER/SURVEILLANCE CAPITALISM

In addition to the glaring backdoor in one of the big techs mobile operating systems mentioned earlier, it is most likely that other bugs and trapdoors may be discovered in the rushed mobile corona virus warning applications. The big techs have grown more powerful during this pandemic. For example, the on-line shopping during this pandemic has lifted the fortunes of one on-line shopping giant at the expense of all the packing material, damage to the enthronement and closing of local stores and boutiques. Also in the few months of the pandemic, the richest people have become yet more so while not contributing anything for spreading correct information or supporting the most dis-advantaged[12]. If nothing is done, the habit formed during this pandemic and the convenience of using the mobile applications to get all one needs would spell disaster for the local small businesses. The reigning in of the big techs is something all states should do. From their history the action taken by the USA's government is sterile. It took decades for them to bring anti-monopoly action of breaking up the telecommunication monopoly. If one looks at the current picture, the units that were spun off have been re-acquired by AT&T and Verizon. The monopoly has been

replaced by a duo-poly. USA's current charade of acting against the big-techs is most likely going to be theatre.

India has recently banned 60 Chinese applications on its mobile system after the latter's purported aggression in the Himalayas. Countries should also start banning the USA's tech giants and prevent them from participating in local grown replacement or fledgling rivals. Cases such as Walmart acquiring 77% of the Indian on-line retailers Flipkart should have been examined by the locals[89]. The success shown against Facebook initiative of taking over the world, under the guise of internet being a basic right[34]: in reality what was being offered is a set of services dictated and dominated by them. This is something similar to the news service being offered by some mobile systems wherein they keep half the revenue and the crumbs are given back to the producers of the news. No wonder some of them were not interested and big techs are labelled free-riders[83].

The replacement for the tech giants should consist of regional and local systems which would safeguard their citizens data, have local control and reduce pollution. A number of open source OSN are available for their implementation; even though they may not have all the bells and whistles: after all the existing systems did not have them at the start. All of this requires political determination and leadership, local talent and local funding and nurturing. It has been reported that UK, in-spite of the fact that it was the first to come up with a test for corona virus, failed to develop their own tracing applications for the lack of capacity and know-how. They had to seek help from the makers of mobile phone operating systems[17, 69]. It appears that the politicians are happy to show that they are doing something by turning to high-tech solutions which are full of bugs, privacy violations and exploitation while need for rapid testing is neglected. It also shows, the dependence on this duo-poly and a reason why independent states must step in and be ready for the next pandemic! The biggest hurdle was never the problem with manual tracing but with poor capacity for doing the tests on suspected cases.

The habit of wearing masks, if it takes hold, would reduce the spread of colds and flu and would be a benefit of Covid-19 pandemic, especially in the western hemisphere[25]. One hopes that a decentralized mobile phone tracking system with distributed data centres, under control of independent responsible authorities, would evolve and be open source. A mechanism should be in place so that the data would be deleted once not required. Any data that is to be used by academic research would have to be done using a public protocol to avoid a repeat of a fiasco such as Cambridge Analytica. The safeguard of the privacy of personal data should be enshrined in the legislation of progressive states[3, 29, 47].

Acknowledgement

The author likes to acknowledge the valuable discussions with and the contribution of Drew Desai (Univ. of Ottawa) and Sheila Desai(BytePress); also, many researchers and journalists cited and perhaps missed; these have been valuable in preparing this article.

REFERENCES

[1] Amnesty International, June 20, 2020, NSO Group spyware used against Moroccan journalist days after company pledged to respect human rights, https://www.amnesty.org/en/latest/news/2020/06/nso-spyware-used-against-moroccan-journalist/
[2] Apuzzo, Matt, Gebrekidan, Selam , Kirkpatrick, David D., How the World Missed Covid-19's Silent Spread, NY Times, June 27, 2020, https://www.nytimes.com/2020/06/27/world/europe/coronavirus-spread-asymptomatic.html
[3] Amnesty International, Joint civil society statement: States use of digital surveillance technologies to fight pandemic must respect, human rights 2 April 2020, https://www.amnesty.org/download/Documents/POL3020812020ENGLISH.pdf
[4] Desai, Bipin C., Navigation Issues Workshop, First World Wide Web Conference, Geneva, May 28, 1994, http://users.encs.concordia.ca/ bcdesai/web-publ/navigation-issues.html
[5] Bipin C. Desai, Pinkerton, Brian, Web—wide Indexing/Semantic Header or Cover Page, Third International World Wide Web Conference, April 10, 1995, http://users.encs.concordia.ca/ bcdesai/web-publ/www3-wrkA/www3-wrkA-proc.pdf
[6] Desai, Bipin C., IoT: Imminent ownership Threat, Proc. IDEAS 2017, July 2017 https://doi.org/10.1145/3105831.3105843
[7] Desai, Bipin C, Privacy in the age of information (and algorithms), Proc. IDEAS '19, June 2019, https://doi.org/10.1145/3331076.3331089
[8] Personal communications, March-June 2020 Residents and new arrivals held prisoners in their own homes/hotels with family members isolated from each other.
[9] Bhambra, Gurminder K, A Statue Was Toppled. Can We Finally Talk About the British Empire?, NY Times, June 12, 2020, https://www.nytimes.com/2020/06/12/opinion/edward-colston-statue-racism.html
[10] Brewster, Jack A Timeline Of The COVID-19 Wuhan Lab Origin Theory, Forbes, May 20, 2020, https://www.forbes.com/sites/jackbrewster/2020/05/10/a-timeline-of-the-covid-19-wuhan-lab-origin-theory/
[11] Browning, Kellen: Twitch Suspends Trump's Channel for 'Hateful Conduct' , June 29, 2020, NY Times, https://www.nytimes.com/2020/06/29/technology/twitch-trump.html
[12] Collins, Chuck In a pandemic, billionaires are richer than ever. Why aren't they giving more? The Guardian, 3 Aug 2020 https://www.theguardian.com/commentisfree/2020/aug/03/billionaires-pandemic-giving-super-rich
[13] Canadian Civil Liberties Association files lawsuit over Sidewalk Labs project, CBC, Apr 16, 2019, https://www.cbc.ca/news/canada/toronto/ccla-sidewalk-labs-lawsuit-1.5100184
[14] Chu, Derek et al.: Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis, The Lancet, June 2020, Citizen Lab, https://doi.org/10.1016/S0140-6736(20)31142
[15] Cecco, Leyland: Google affiliate Sidewalk Labs abruptly abandons Toronto smart city project, The Guardian, 7 May 2020, https://www.theguardian.com/technology/2020/may/07/google-sidewalk-labs-toronto-smart-city-abandoned
[16] NSO Group/Q Cyber Technologies Over One Hundred New Abuse Cases, Citizen Lab, October 2019, https://citizenlab.ca/2019/10/nso-q-cyber-technologies-100-new-abuse-cases/
[17] Clark, Pilita; Cookson, Clive; Hughes, Laura: How the UK got coronavirus testing wrong Finanacial Times, March 27, 2020 https://www.ft.com/content/fa747fbd-c19e-4bac-9c37-d46afc9393fb
[18] Contact Tracing, https://en.wikipedia.org/wiki/Contact_tracing
[19] Contact tracing, http://www.bccdc.ca/health-info/diseases-conditions/covid-19/self-isolation/contact-tracing
[20] Decentralized Privacy-Preserving Proximity Tracing, https://en.wikipedia.org/wiki/Decentralized_Privacy-Preserving_Proximity_Tracing
[21] Diamond, Stephen: Statement From Waterfront Toronto Board Chair, May 7, 2020, https://quaysideto.ca/wp-content/uploads/2020/05/Waterfront-Toronto-Statement-May-7-2020.pdf
[22] The Editors of Encyclopedia Britannica, Influenza pandemic of 1918–19,
[23] Foer, Franklin, What Big Tech Wants Out of the Pandemic, TThe Atlantic July/August 2020 https://www.theatlantic.com/magazine/archive/2020/07/big-tech-pandemic-power-grab
[24] Frank, Thomas: The Pessimistic Style in American Politics, Harper, May 2020, https://harpers.org/archive/2020/05/how-the-anti-populists-stopped-bernie-sanders/
[25] Gandhi, Monica; Beyrer Chris; Goosby, Eric: Masks Reduce Viral Inoculum of SARS-CoV2 Journal of General Internal Medicine (August 2020)
[26] Understanding Mobile Apps, https://www.consumer.ftc.gov/articles/0018-understanding-mobile-apps
[27] Genomic epidemiology of hCoV-19, GISAID, https://www.gisaid.org/epiflu-applications/next-hcov-19-app/
[28] Guarino, Ben; Janes, Chelsea; Cha, Ariana Eunjung: Spate of new research supports wearing masks to control coronavirus spread, Wash Post, June, 13,

2020, https://www.washingtonpost.com/health/2020/06/13/spate-new-research-supports-wearing-masks-control-coronavirus-spread/
[29] Guinan, Joe; O'Neill, Martin: Only bold state intervention will save us from a future owned by corporate giants, The Guardian, 6 Jul 2020, https://www.theguardian.com/commentisfree/2020/jul/06/state-intervention-amazon-recovery-covid-19
[30] Granville, Kevin: Facebook and Cambridge Analytica: What You Need to Know as Fallout Widens, NY Times, March 19, 2018, https://www.nytimes.com/2018/03/19/technology/facebook-cambridge-analytica-explained.html
[31] Glavin, Terry: Footprints of the corona-virus: How it came to Canada and went around the world, Maclean's, May 7, 2020, https://www.macleans.ca/society/health/footprints-of-the-coronavirus-how-it-came-to-canada-and-went-around-the-world/
[32] Hutchins, Aaron: Contact tracers are the new front line against COVID-19, Macleans, June 26, 2020, https://www.macleans.ca/society/health/contract-tracers-canadas-coronavirus-crisis/
[33] Editors - History.com, Pandemics that changed history, April 2020, https://www.history.com/topics/middle-ages/pandemics-timeline
[34] Hempel, Jessy: What Happened to Facebook's Grand Plan to Wire the World?, Wired, 05.17.2018, https://www.wired.com/story/what-happened-to-facebooks-grand-plan-to-wire-the-world/
[35] Horowitz, Jeff, Setharamman, Deepa, Facebook Executives Shut Down Efforts to Make the Site Less Divisive, Wall Street Journal, May 26, 2020, https://www.wsj.com/articles/facebook-knows-it-encourages-division-top-executives-nixed-solutions-11590507499
[36] A Failure Of Initiative, Report by the Select Bipartisan Committee to Investigate the Preparation for and Response to Hurricane Katrina, February 15, 2006. https://www.nrc.gov/docs/ML1209/ ML12093A081.pdf
[37] Hogan Libbyin; Safi, Michael: Revealed: Facebook hate speech exploded in Myanmar during Rohingya crisis, The Guardian, April 3, 2018, https://www.theguardian.com/world/2018/apr/03/revealed- facebook -hate-speech-exploded-in-myanmar-during-rohingya-crisis
[38] Hackett, Robert, Sidewalk Labs' Toronto project was dead on arrival, Fortune, May 13, 2020, https://fortune.com/2020/05/13/sidewalk-labs-toronto-waterfront-quayside-smart-city/
[39] HT Correspondent Pegasus creator NSO Group has a Covid-19 software: Why you should be worried, 07 Apr 2020, https://tech.hindustantimes.com/tech/news/pegasus-creator-nso-group-has-a-covid-19-software-why-you-should-be-worried-story-SMsYXKFoz9OPG2VBIKrA8L.html
[40] Harari Yuval Noah: Why Technology Favors Tyranny, The Atlantic, October 2018, https://www.theatlantic.com/magazine/archive/2018/10/yuval-noah-harari-technology-tyranny/568330/568330/
[41] Kilbourne Edwin D.: Influenza Pandemics of the 20th Century, Emerg Infect Dis. 2006 Jan; 12(1): 9–14. https://dx.doi.org/10.3201/eid1201.051254
[42] Kleeman, Jenny: My son was shaking, trying not to go online': how the gambling industry got its claws into children, The Guardain, July 11, 2020, https://www.theguardian.com/lifeandstyle/2020/jul/11/my-son-would-be-shaking-trying-not-to-go-online-how-the-gambling-industry-got-its-claws-into-kids
[43] Klein, Naomi: Disaster capitalism: how to make money out of misery, The Guardain, 30 Aug 2006, https://www.theguardian.com/commentisfree/2006/aug/30/comment.hurricanekatrina
[44] Klein, Naomi: Under Cover of Mass Death, Andrew Cuomo Calls in the Billionaires to Build a High-Tech Dystopia, The Intercept, May 8 2020, https://theintercept.com/2020/05/08/andrew-cuomo-eric-schmidt-coronavirus-tech-shock-doctrine/
[45] Klein, Naomi: How big tech plans to profit from the pandemic, The Guardian, May 13, 2020, https://www.theguardian.com/news/2020/may/13/naomi-klein-how-big-tech-plans-to-profit-from-coronavirus-pandemic
[46] Lee Fang, Lee: FBI Expands Ability to Collect Cellphone Location Data, Monitor Social Media, Recent Contracts Show The Intercept, June 24 2020, Thttps://theintercept.com/2020/06/24/fbi-surveillance-social-media-cellphone-dataminr-venntel/
[47] Lomas, Natasha: An EU coalition of techies is backing a 'privacy-preserving' standard for COVID-19 contacts tracing, April, 1, 2020, https://techcrunch.com/2020/04/01/an-eu-coalition-of-techies-is-backing-a-privacy-preserving-standard-for-covid-19-contacts-tracing/
[48] McKibben, Bill: What Facebook and the Oil Industry Have in Common, The New Yorker, July 1, 2020, https://www.newyorker.com/news/annals-of-a-warming-planet/what-facebook-and-the-oil-industry-have-in-common
[49] Manjoo, Farhad, Tech's Frightful Five: They've Got Us, NY Times, May 10, 2017, https://www.nytimes.com/2017/05/10/technology/techs-frightful-five-theyve-got-us.html
[50] Malik, Kenan: For all its sophistication, AI isn't fit to make life-or-death decisions, The Guardian, May 16, 2020, https://www.theguardian.com/commentisfree/2020/may/16/for-all-its-sophistication-ai-isnt-fit-to-make-life-or-death-decisions-for-us
[51] McKie, Robin, Tapper, James, Savage, Michael, Boris Johnson told to dump rhetoric and plan for new Covid wave, The Guardian, June, 6, 2020, https://www.theguardian.com/world/2020/jun/06/prime-minister-told-to-dump-rhetoric-and-plan-for-new-covid-wave
[52] Mozur, Paul: A Genocide Incited on Facebook, With Posts From Myanmar's Military, NYTimes, Oct 2018, https://www.nytimes.com/2018/10/15/technology/myanmar-facebook-genocide.html
[53] Morales Steve No touching, jokes or pirouettes: when Canadians offended royal protocol. Global News, November 25, 2015, https://globalnews.ca/news/2362049/no-touching-jokes-or-pirouettes-when-canadians-offended-royal-protocol/
[54] NCSAI Commissioners, Accessed June, 2020, https://www.nscai.gov/about/commissioners
[55] Singer, Natasha, Google Promises Privacy With Virus App but Can Still Collect Location Data, NY Times, July 20, 2020, https://www.nytimes.com/2020/07/20/technology/google-covid-tracker-app.html
[56] NSO Group/Q Cyber Technologies Over One Hundred New Abuse Cases. Accessed June, 2020, https://citizenlab.ca/2019/10/nso-q-cyber-technologies-100-new-abuse-cases/
[57] Ovide, Shira, Conviction in the Philippines Reveals Facebook's Dangers, NY Times, June 16, 2020, https://www.nytimes.com/2020/06/16/technology/facebook-philippines.html
[58] The birth of IBM PC, https://www.ibm.com/ibm/history/exhibits/pc25/pc25_birth.html
[59] Perlroth, Nicole: WhatsApp Says Israeli Firm Used Its App in Spy Program, Oct. 29, 2019, hhttps://www.nytimes.com/2019/10/29/technology/whatsapp-nso-lawsuit.html
[60] Pan-European Privacy-Preserving Proximity Tracing, https://en.wikipedia.org/wiki/Pan-European_Privacy-Preserving_Proximity_Tracing
[61] Request for Proposals Innovation and Funding Partner for the Quayside Development Opportunity https://quaysideto.ca/wp-content/uploads/2019/04/Waterfront-Toronto-Request-for-Proposals-March-17-2017.pdf
[62] Rivero, Daniel, Everything that's wrong with police training in one chart, Splinter News, Aug. 20. 2015, https://splinternews.com/everything-thats-wrong-with-police-training-in-one-char-1793850106
[63] Rogers, Kara: 1957 flu pandemic, https://www.britannica.com/event/1957-flu-pandemic
[64] Rogers, Kara: 1968 flu pandemic, https://www.britannica.com/event/1968-flu-pandemic
[65] Roose, Kevin: Social Media Giants Support Racial Justice. Their Products Undermine It, NY Times, 19, June 2020, https://www.nytimes.com/2020/06/19/technology/facebook-youtube-twitter-black-lives-matter.html
[66] Rich, Motoko: Is the Secret to Japan's Virus Success Right in Front of Its Face?, NYTimes, June 6,2020, https://www.nytimes.com/2020/06/06/world/asia/japan-coronavirus-masks.html
[67] Rich, Motoko, Tokyo in a State of Emergency, Yet Still Having Drinks at a Bar, NYTimes, April 19, 2020, https://www.nytimes.com/2020/04/19/world/asia/tokyo-japan-coronavirus.html
[68] Rutstein, David D.: The Influenza Epidemic, Harpers, Aug. 1957, https://harpers.org/archive/1957/08/the-influenza-epidemic/
[69] Satariano, Adam; Mueller, Benjamin:Mueller, Benjamin: Britain Didn't Want Silicon Valley's Help on a Tracing App. Now It Does, NY Times, June 18, 2020 https://www.nytimes.com/2020/06/18/business/britain-contact-tracing-app.html
[70] Sang-Hun, Choe; Krolik, Aaron; Zhong, Raymond; Singer, Natasha: Major Security Flaws Found in South Korea Quarantine App NYTimes, July 21, 2020, https://www.nytimes.com/2020/07/21/technology/korea-coronavirus-app-security.html
[71] Schmidt, Eric, I Used to Run Google. Silicon Valley Could Lose to China., NY Times, Feb. 27, 2020, https://www.nytimes.com/2020/02/27/opinion/eric-schmidt-ai-china.html
[72] Satter, Raphael: WhatsApp takes step toward winning spyware lawsuit after Israeli company no-show, Reuters, March 3, 2020, https://www.reuters.com/article/us-whatsapp-court/whatsapp-takes-step-toward-winning-spyware-lawsuit-after-israeli-company-no-show-idUSKBN20R02M
[73] Sidewalk Lab's Proposal: Master Innovation and Development Plan, June 17, 2019 https://quaysideto.ca/sidewalk-labs-proposal-master-innovation-and-development-plan/
[74] Strochlic, Nina; Champine, Riley D.: How some cities 'flattened the curve' during the 1918 flu pandemic, National Geographic, March 27,

2020, https://www.nationalgeographic.com/history/2020/03/how-cities-flattened-curve-1918-spanish-flu-pandemic-coronavirus/
[75] Senate Report, The Phoenix Pay Problem: Working Towards a Solution (PDF). Standing Senate Committee on National Finance, Report of the Standing Senate Committee on National Finance. Ottawa, Ontario. July 31, 2018. p. 34. Retrieved May 21, 2019., https://sencanada.ca/content/sen/committee/421/NFFN/Reports/NFFN_Phoenix_Report_32_WEB_e.pdf
[76] Silva, Shiroma, How map hacks and buttocks helped Taiwan fight Covid-19, BBC Click 7 June 2020, https://www.bbc.com/news/technology-52883838
[77] Shaikh, Shadma: Why mobile apps require access to your data and device tools, India Times, Aug 05, 2019, https://economictimes.indiatimes.com/small-biz/security-tech/technology/why-mobile-apps-require-access-to-your-dataand-device-tools/articleshow/52138161.cms
[78] Tusikov, Natasha, Sidewalk Toronto's master plan raises urgent concerns about data and privacy, The Conversations, July, 30 2019, https://theconversation.com/sidewalk-torontos-master-plan-raises-urgent-concerns-about-data-and-privacy-121025
[79] Tonar, Remington; Talton, Ellis: Why Sidewalk Labs' Toronto Plan Is Flawed, Forbes, Sep 26, 2019, https://www.forbes.com/sites/ellistalton/2019/09/26/why-sidewalk-labs-toronto-plan-is-flawed/#36788aa96bda
[80] Taylor, Diane: Serco wins Covid-19 test-and-trace contract despite £1m fine, The Guardian, June 6. 2020, https://www.theguardian.com/world/2020/jun/06/serco-wins-covid-19-test-and-trace-contract-despite-1m-fine
[81] Schrader, Stuart: Yes, American police act like occupying armies. They literally studied their tactics , The Guardian, June 8, 2020, https://www.theguardian.com/commentisfree/2020/jun/08/yes-american-police-act-like-occupying-armies-they-literally-studied-their-tactics
[82] Videotelephony, https://en.wikipedia.org/wiki/Videotelephony
[83] Vega, Nicolas: New York Times pulls out of Apple News partnership , NY Post, June 29,. 2020. https://nypost.com/2020/06/29/new-york-times-pulls-out-of-apple-news-partnership/
[84] Walton, Alice G.: Face Masks May Be The Key Determinant Of The Covid-19 Curve, Study Suggests, Forbes, June 13, 2020, https://www.forbes.com/sites/alicegwalton/2020/06/13/face-masks-may-be-the-key-determinant-of-the-covid-19-curve-study-suggests/
[85] Wylie Bianca: Searching for the Smart City's Democratic Future, August 13, 2018 https://www.cigionline.org/articles/searching-smart-citys-democratic-future
[86] Walsh, Bryan, Covid-19: The history of pandemics, BBC, March 25, 2020, https://www.bbc.com/future/article/20200325-covid-19-the-history-of-pandemics
[87] Winder, Davey: U.S. Government Issues Critical Windows 10 'Update Now' Alert, Forbes, Jan 15, 2020, https://www.forbes.com/sites/daveywinder/2020/01/15/us-government-issues-critical-windows-10-update-now-alert
[88] Wikipedia, COVID-19 apps, https://en.wikipedia.org/wiki/COVID-19_apps
[89] Flipkart, https://en.wikipedia.org/wiki/Flipkart
[90] MS-DOS, https://en.wikipedia.org/wiki/MS-DOS
[91] Phoenix pay system, https://en.wikipedia.org/wiki/Phoenix_pay_system
[92] List of countries by smartphone penetration, https://en.wikipedia.org/wiki/List_of_countries_by_smartphone_penetration
[93] WannaCry ransomware attack https://en.wikipedia.org/wiki/WannaCry_ransomware_attack
[94] COVID-19 pandemic by country and territory, https://en.wikipedia.org/wiki/COVID-19_pandemic_by_country_and_territory
[95] Zhang, Renyi; Li, Yixin, Zhang; Annie L.; Wng, Yuan; Molina, Mario J.: Identifying airborne transmission as the dominant route for the spread of COVID-19, PNAS, June 11, 2020 https://doi.org/10.1073/pnas.2009637117

Avoiding Blocking by Scheduling Transactions using Quantum Annealing

Tim Bittner
Sven Groppe
tim.bittner96@gmx.de
groppe@ifis.uni-luebeck.de
Institute of Information Systems (IFIS), University of Lübeck
Lübeck

ABSTRACT

Quantum annealers are a special kind of quantum computers for solving optimization problems. In this paper, we investigate the benefits of quantum annealers in the field of transaction synchronization. In particular, we show how transactions using the 2-phase-locking protocol can be optimally distributed to any number of available machines to reduce transaction waiting times. Therefore an instance of the problem will be transformed into a formula that is accepted by quantum annealers. In an experimental evaluation, the runtime on a quantum annealer outperforms the runtime of traditional algorithms to solve combinatorial problems like simulated annealing already for small problem sizes.

CCS CONCEPTS

• **Information systems** → **Data management systems**; **Database transaction processing**.

KEYWORDS

Quantum computing, quantum annealing, transaction processing, synchronization, 2-phase-locking, database, schedule, D-Wave

ACM Reference Format:
Tim Bittner and Sven Groppe. 2020. Avoiding Blocking by Scheduling Transactions using Quantum Annealing. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410593

1 MOTIVATION

In the last decade quantum computing including special forms like quantum annealing solving quadratic unconstrained binary optimization (QUBO) problems [2, 12] has become more and more popular and the range of applications wider and wider [15, 17, 20, 21], leading to a lot of time and money being spent on research in this area. Many scientific papers deal with the comparison of standard algorithms and quantum algorithms [7] and quite a few of them belong to the field of optimization problems [22, 24, 25]. There are only few contributions dealing with optimizing database problems with quantum computers like query optimization [24].

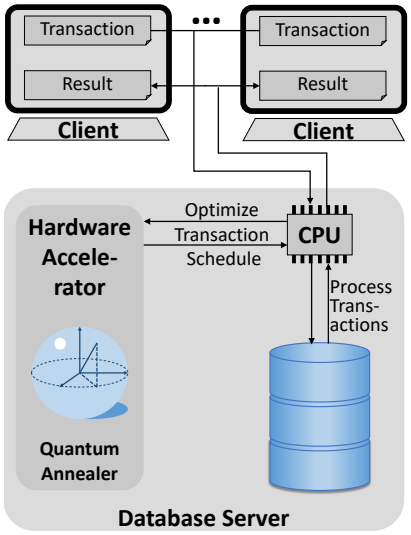


Figure 1: Hardware accelerating transaction schedules via quantum annealing

The field of transaction synchronization includes many combinatorial problems and hence offers various problems and certainly plenty of potential for optimizations by quantum computers. This work focuses in particular on the optimization of the 2-phase-locking protocol for transactions, which prevents conflicts between transactions by *preclaiming* all necessary locks at begin of transaction and avoids cascading aborts by holding all locks until the end of transaction, but forces waiting times due to the locks. Our contribution deals with avoiding waiting times and finding an optimal execution sequence. Thereby a solution approach for quantum annealers used as hardware accelerator for database tasks [14] is derived (see Figure 1), which are designed to solve hard optimization problems [1]. Since inputs of quantum annealers cannot be arbitrary, some preprocessing is required to transform given problems in such a way that processing by a quantum annealer is possible. After preprocessing, quantum annealers solve the problem in constant time in contrast to other hardware accelerators [13]. Our work is inspired by work on job shop scheduling problems (being among the hardest combinatorial optimization problems [11]) for quantum

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410593

computers [25], but transaction scheduling additionally needs to consider conflicts between transactions.

Our main contributions are

- proposing a formula runnable on quantum annealers for an optimal scheduling of transactions synchronized by 2-phase-locking protocol with preclaiming of all locks at begin of transaction and holding all locks until end of transaction,
- a complexity analysis of the preprocessing time and the number of required qubits, and
- an extensive experimental evaluation comparing the runtimes of quantum annealers with the simulated annealing algorithm.

We introduce the basics about transactions, transaction management and quantum computers in Section 2. Section 3 covers the main part of the work, the formal model for the problem and the transformation from this model into a formula suitable for a quantum annealer. We provide a short insight into the implementation and the used software in Section 4. We estimate the number of required qubits, and analyze the complexity of preprocessing and experiments on a quantum annealer in Section 5. We finally summarize and provide a small outlook for future work in Section 6.

2 BASICS

This section is dedicated to the basics of transaction management (see Section 2.1) and quantum computers (see Section 2.2) with special focus on quantum annealing.

2.1 Transaction Management

A **transaction** $t =< s_1, ..., s_n >$ of length n is a series of operations s_i, carried out by a single user or application program, which reads or updates the contents of the database [8]. Each operation $s_i = a_i(e_i)$ consists of the type of access $a_i \in \{r, w\}$, where r represents a read access and w a write access, and the object e_i to be accessed.

For example, the transaction $t =< r(A), w(A), r(B), w(B) >$ is of length $|t| = 4$.

2.1.1 Conflicts between Transactions: Since transactions often need to access a database at the same time and thereby often reference the same objects, transactions in a database system are required to fulfill the so-called ACID properties [8]. The focus here is on the fulfillment of the (I)solation property, which guarantees to avoid problems of unsynchronized parallel execution of several transactions and requires concurrently executed transactions to not influence each other.

2.1.2 Conflict Management: To ensure the isolation property, conflicts between transactions must be dealt with. The simplest strategy to deal with conflicts would be to execute the transactions serially. Since this is too slow for operational systems and also many transactions do not conflict with each other at all, in practice transactions are executed in parallel. There are various approaches to dealing with the conflicts that arise, one of them is the use of locks. For this purpose, each transaction acquires a lock for an object before access and releases it after access, thus preventing concurrent access to an object.

The **two-phase-locking protocol** is a locking protocol that requires each transaction consisting of two subsequent phases, the locking phase and the release phase. During the locking phase, the transaction may acquire locks but not release them, whereas the release phase requires the release of previously required locks. If a transaction has released a lock, it may not acquire any new ones. There is a distinction between *read locks* (also called *shared locks*) and *write locks* (also called *exclusive locks*). Variants of the protocol include the conservative two-phase-locking protocol and the strict two-phase-locking protocol. The conservative variant (*preclaiming*) requires that all locks that are required during a transaction are acquired before the transaction is started.[1] The strict variant holds all locks until the end of a transaction, which avoids *cascading aborts* occurring in case of so called *dirty reads* of objects written by transactions, which are later aborted resulting in an abort of the current transaction as well. In this contribution, we consider the combination of the conservative and the strict two-phase-locking protocol in our transaction model, which results in a serial execution of all transactions that access the same objects.

2.2 Quantum computer

Quantum computers are computers that are not based on classical mechanics, instead they exploit the effects of quantum mechanics.

2.2.1 Basic idea: Quantum mechanics describes the states and behavior of particles that are smaller than the size of an atom and do not follow the laws of classical physics. At this scale, there occur effects that the quantum computer makes use of, especially the principle of superposition and that of quantum entanglement. The quantum computer uses qubits, which can take on 2 states simultaneously due to the principle of superposition. While a bit can assume the state 0 or 1, a qubit assumes the states 0 and 1 simultaneously. If a measurement of the state is made, the qubit changes to one of the two states. Both states have relative probabilities with which they are assumed in the measurement. The principle of quantum entanglement enables the mutual influence of qubits, since entangled qubits mutually influence their probabilities. Imagining the principle of superposition as a special form of parallel computing opens up a new world of computation beyond polynomial time and allows in theory an exponential speedup compared to classical computers.

2.2.2 Quantum computing. finds desired solutions of a problem by clever manipulation of single qubits as well as entangled qubits. For the purpose of manipulating qubits, gates provide elementary operations on one or two qubits. For example, the Hadamard-Gate puts one qubit into superposition and a Controlled-NOT(CNOT)-Gate inverts a second qubit depending on the first [6]. A quantum computer is thus able to execute quantum algorithms like Shor's algorithm for factorizing large numbers [21] or Grover's algorithm for searching in huge unsorted databases [15, 16].

2.2.3 Quantum Annealers. are a special form of quantum computers [10, 18], which are designed to solve hard optimization problems, but cannot execute quantum algorithms like Shor's algorithm. Quantum annealers find the global minimum of a given objective function by transforming a simplified objective function

[1] Please note that for those transactions, for which the required locks are not known before processing, the required locks can be determined by an additional phase before transaction processing. The contribution in [23] describes such an approach which can be also applied in our scenario.

with known global minimum into the objective function of interest, so that the quantum annealer always remains in the state of the global minimum, which represents the desired result. The most widely-known quantum annealers are manufactured by the canadian company D-Wave [1], which produced the first commercial quantum computers.

Input format: Quantum annealers have very limited input formats, such that all problems have to be transformed into this input format before being solved and transformed back after annealing. Quantum annealers solve quadratic unconstrained binary optimization (QUBO) problems [2, 12], which belong to the class of NP-hard problems. A QUBO-problem is defined by N weighted binary variables $X_1, ..., X_N \in \{0, 1\}$, either as linear or quadratic term:

$$\sum_{0<i\leq N} w_i X_i + \sum_{i\leq j\leq N} w_{ij} X_i X_j, \text{ where } w_i, w_{ij} \in \mathbb{R}$$

The quantum annealer transforms these weightings into energy levels of single qubits or between two qubits, and by minimizing the energy level the quantum annealer finds a minimization of the objective function, in other words a variable assignment for the QUBO-problem. The idea of quantum annealing is very close to the simulated annealing approach [19] on traditional computers (see Figure 2).

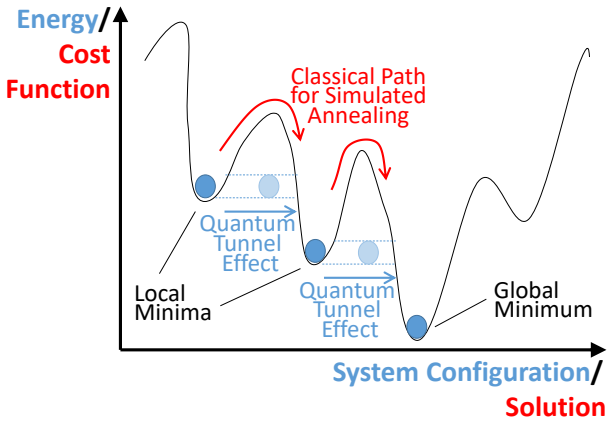


Figure 2: Simulated annealing (sa) [19] versus quantum annealing (qa) to reach a global minimum (simplified discussion): Quantum tunneling driven by an external magnetic field "passes" high energy/cost function "peaks" instead of climbing them as in sa. Larger tunnels can be passed (qa) and bigger peaks climbed (sa) with a high magnetic field (qa) and a high temperature (sa). Then lower magnetic fields (qa) and lower temperatures (sa) hinders to get out of a valley of the energy (qa) and cost function (sa) respectively with a global minimum. As long as the function fits into the system configuration size, qa determines the minimum in constant time in contrast to sa, which takes longer times for more complex functions.

3 MODELLING

In this section, we describe the formal model of instances of the considered scheduling problem (see Section 3.1) and afterwards formulate the scheduling problem as QUBO-problems (see Section 3.2) for valid solutions (see Section 3.3) and optimal solutions (see Section 3.3.1), and optimize the total solution further (see Section 3.3.2).

3.1 Formal model

The goal of this optimization is to reduce, or in the best case completely eliminate, the waiting times of transactions that use the two-phase-locking protocol. A transaction must wait whenever it requests a lock, but this lock is held by another transaction. In this case, one transaction blocks the other and only one of the two can be executed at a time. For simplicity of presentation, we consider the conservative and strict two-phase-locking protocol[2]. We assume that k machines are available for processing n transactions. Thus k transactions can run parallel to each other, if they do not block each other. We hence consider the scheduling problem, where n transactions have to be distributed to k machines, taking into account that two transactions blocking each other are never executed simultaneously. Thereby the maximum execution time R (see Section 3.3.3) should be minimized over all machines. An instance of this scheduling problem consists of a set T of transactions with $|T| = n$, a set M of machines with $|M| = k$ (see Figure 6) and a set $O \subseteq T \times T$ of blocking transactions (see Figure 5). Each transaction $t_i \in T$ has a certain length l_i (see Figure 3) and thus an upper bound $r_i = R - l_i$ for its start time (see Figure 4).

3.1.1 Running Examples. This paper contains two running examples:

Example E_1: The first running example E_1 deals with a rather complex scenario and contains 8 transactions for illustrating the model in figures 3, 4, 5, 6, 7 and 9.

Example E_2: The second running example E_2 contains a simpler configuration of transactions, but is especially designed for illustrating the generated formula to be minimized by a quantum annealer. In E_2, three transactions are to be distributed on two machines. The first transaction has a length of 2, the other two a length of 1. Furthermore, the second and third transactions are blocking each other in their execution. We use $R = 2$ for the maximum execution time (see Section 3.3.3 for determining R).

Overall we have the following configuration for the running example E_2:

$$R = 2$$
$$T = \{t_1, t_2, t_3\} \; with \; |T| = 3$$
$$M = \{m_1, m_2\} \; with \; |M| = 2$$
$$O = \{(t_2, t_3)\}$$
$$l_1 = 2, l_2 = 1, l_3 = 1$$
$$r_1 = 0, r_2 = 1, r_3 = 1$$

[2] As discussed in Section 2.1.2 and by using the approach in [23], transactions can be also processed, which do not have a fixed set of required locks, by introducing an additional processing phase before transaction start.

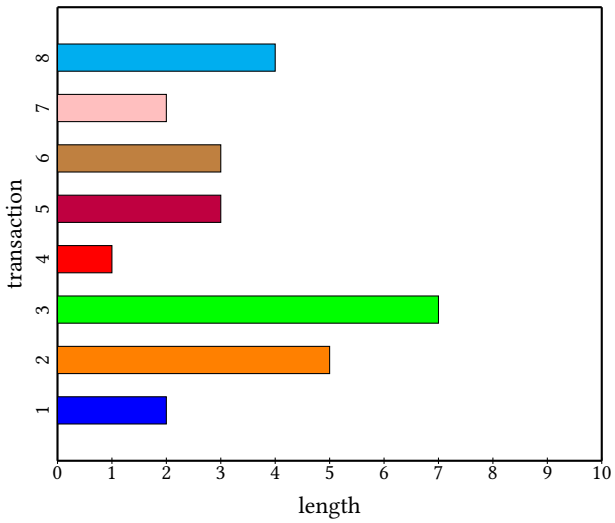


Figure 3: $n = 8$ transactions with their respective lengths (Example E_1)

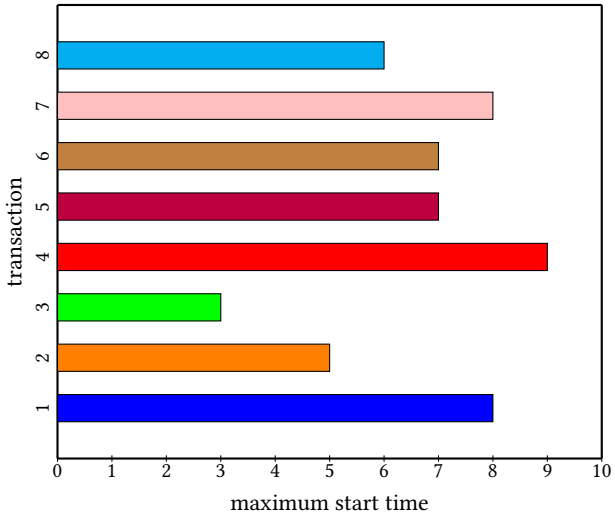


Figure 4: $n = 8$ transactions with their respective maximum start times (Example E_1)

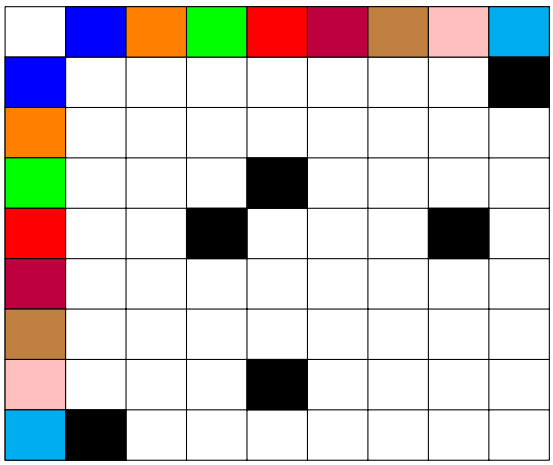


Figure 5: Black fields indicate blocking transactions (Example E_1)

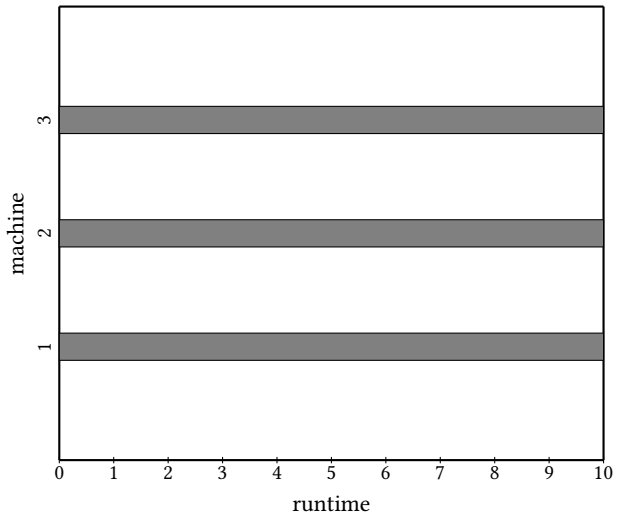


Figure 6: Transactions are scheduled on three machines, the maximum runtime here is $R = 10$ (Example E_1)

3.2 Formulation as QUBO-problem

A QUBO-problem consists of binary variables that occur weighted in linear or quadratic terms. The binary variables

$$X_{i,j,s} \text{ for } 1 \leq i \leq n, 1 \leq j \leq k, 0 \leq s \leq r_i$$

contain the value 1 if transaction t_i is started at time s on machine m_j, otherwise 0. For a valid schedule n of the variables must hold the value 1 and all others 0, so that for each transaction the respective start time is expressed. The solution to this problem is the distribution of the transactions to the different machines (see Figure 7).

3.3 Valid solution

Constraints for a valid schedule are now formulated in such a way that the resulting formula takes on high values whenever the constraints for validity are not satisfied and low values whenever they are satisfied. Three constraints must be formulated so that minimizing the formula guarantees a valid schedule:

A: Each transaction starts exactly once,
B: two or more transactions cannot be executed at the same time on the same machine, and
C: transactions that block each other cannot be executed at the same time.

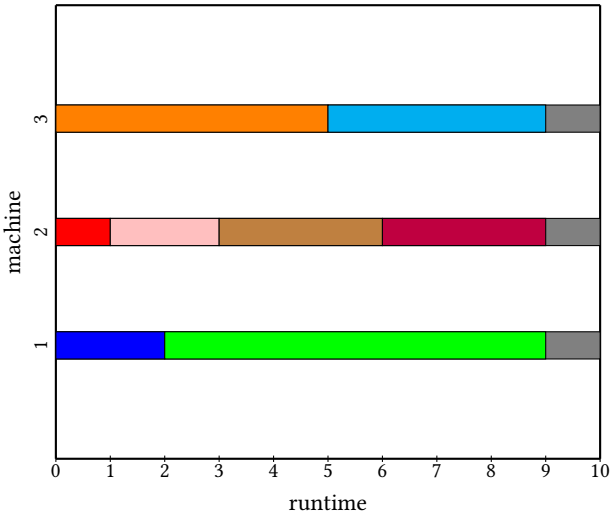


Figure 7: Binary variables set for this schedule: $X_{110}, X_{312}, X_{420}, X_{721}, X_{623}, X_{526}, X_{230}, X_{835}$ (Example E_1)

To ensure that each transaction starts exactly once, it may only start once across all machines and possible start times:

$$A = \underbrace{\sum_{i=1}^{n}}_{\text{transactions}} \Big(\underbrace{\sum_{j=1}^{k}}_{\text{machines}} \underbrace{\sum_{s=0}^{r_i}}_{\text{start times}} X_{i,j,s} - 1 \Big)^2$$

For the example configuration E_2 of Section 3.1.1:

$$\begin{aligned} A = {} & (X_{1,1,0} + X_{1,2,0} - 1)^2 \\ & + (X_{2,1,0} + X_{2,1,1} + X_{2,2,0} + X_{2,2,1} - 1)^2 \\ & + (X_{3,1,0} + X_{3,1,1} + X_{3,2,0} + X_{3,2,1} - 1)^2 \end{aligned}$$

A holds the value 0 if each transaction starts exactly once, a higher one otherwise. If for a transaction the rear term in brackets is multiplied with $k \cdot r_i$ variables, only terms with variables appear and offsets of $1 = -1 \cdot -1$. Hence we obtain n offsets for n variables. Since a QUBO-problem consists solely of linear and quadratic terms, the offset has no place there and is ignored. Hence during running the problem on a quantum annealer, A finally retrieves only the value $-n$ and the offset of n is ignored. After the final formula has been minimized and a variable assignment has been found, the offset can of course be added again, but it does not play any role in solving the problem.

To ensure that transactions do not run at the same time on the same machine, the runtime is calculated in the following way using the start time and the length of the transactions:

$$B = \underbrace{\sum_{j=1}^{k}}_{\text{machines}} \overbrace{\sum_{i_1=1}^{n-1}}^{\text{transactions without } t_n} \underbrace{\sum_{s_1=0}^{r_{i_1}}}_{\text{start times}} \overbrace{\sum_{i_2=i_1+1}^{n}}^{\text{remaining transactions}} \underbrace{\sum_{s_2=q}^{p}}_{\text{invalid start times}} X_{i_1,j,s_1} X_{i_2,j,s_2}$$

for $q = \max\{0, s_1 - l_{i_2} + 1\}, p = \min\{s_1 + l_{i_1}, r_{i_2}\}$

For the example configuration E_2 of Section 3.1.1:

$$\begin{aligned} B = {} & X_{1,1,0}X_{2,1,0} + X_{1,1,0}X_{2,1,1} + X_{1,1,0}X_{3,1,0} \\ & + X_{1,1,0}X_{3,1,1} + X_{2,1,0}X_{3,1,0} + X_{2,1,1}X_{3,1,1} \\ & + X_{1,2,0}X_{2,2,0} + X_{1,2,0}X_{2,2,1} + X_{1,2,0}X_{3,2,0} \\ & + X_{1,2,0}X_{3,2,1} + X_{2,2,0}X_{3,2,0} + X_{2,2,1}X_{3,2,1} \end{aligned}$$

If no transactions are running at the same time on the same machine, at most one of the variables pairwise takes the value 1 and B overall takes the value 0. For each machine, transaction and associated start time, all invalid start times of all remaining transactions are calculated and this combination is added to the formula. B reaches the maximum, i.e., exactly the value of all sums of the formula, in the case of pairwise both variables take the value 1. The values q and p delimit the range in which two transactions overlap in their runtimes for certain start times (see Figure 8).

To avoid transactions that block each other being executed at the same time, similar constraints are established as for B:

$$C = \underbrace{\sum_{\{t_{i_1}, t_{i_2}\} \in O}}_{\text{blocking transactions}} \overbrace{\sum_{j_1=1}^{k}}^{\text{machines}} \underbrace{\sum_{s_1=0}^{r_{i_1}}}_{\text{start times}} \overbrace{\sum_{j_2 \in J}}^{\text{remaining machines}} \underbrace{\sum_{s_2=q}^{p}}_{\text{invalid start times}} X_{i_1,j_1,s_1} X_{i_2,j_2,s_2}$$

for $J = \{1, ..., k\} \setminus \{j_1\}$,

$q = \max\{0, s_1 - l_{i_2} + 1\}, p = \min\{s_1 + l_{i_1}, r_{i_2}\}$

For the example configuration E_2 of Section 3.1.1:

$$\begin{aligned} C = {} & X_{2,1,0}X_{3,2,0} + X_{2,1,1}X_{3,2,1} \\ & + X_{2,2,0}X_{3,1,0} + X_{2,2,1}X_{3,1,1} \end{aligned}$$

Similar to B, C takes the value 0, if no transactions that block each other are executed at the same time and so again only one of the variables pairwise takes the value 1. For all pairs of blocking transactions, the different invalid start combinations are determined and added to the formula. C reaches the maximum, i.e., exactly the value of all sums of the formula, in the case of pairwise both variables take the value 1.

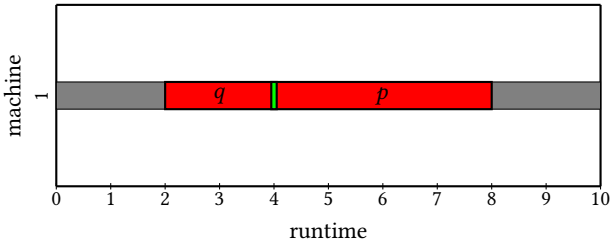


Figure 8: The green line represents a starting time 4 of a transaction of length 4 on a machine. The value q indicates that a second transaction of length 3 must not be started in the red area in front of the green line, because otherwise the runtimes will overlap. The value p expresses the time until when the first transaction runs, so that no other transactions may be started in the second red area either.

3.3.1 Optimal solution. For an optimal solution, the variables are now weighted so that minimizing the formula requires the earliest possible start time for each transaction. Therefore the calculated end time is weighted so that its weight exceeds the sum of all weights of all machines for each smaller end time. This is necessary because otherwise many short transactions that are started early would keep the weight lower than long transactions. As a result, long transactions with a different weight distribution would only be started at the end and thus the actual goal of minimizing the maximum execution time would not be achieved. The weight for an end time $s + l_i$ looks like this (see Figure 9):

$$w_{s+l_i} = \frac{(k+1)^{s+l_i-1}}{(k+1)^R}$$

By dividing by $(k+1)^R$, $w_{s+l_i} \in (0,1)$ applies, which requires that a valid solution is preferred to an optimal one, since the weights in A, B and C always assume the value $|1|$ or higher. This results in the following formula:

$$D = \sum_{i=1}^{n} \sum_{j=1}^{k} \sum_{s=0}^{r_i} w_{s+l_i} X_{i,j,s}$$

For the example configuration E_2 of Section 3.1.1:

$$\begin{aligned} D = & \frac{3}{9}X_{1,1,0} + \frac{3}{9}X_{1,2,0} \\ & + \frac{1}{9}X_{2,1,0} + \frac{3}{9}X_{2,1,1} + \frac{1}{9}X_{2,2,0} + \frac{3}{9}X_{2,2,1} \\ & + \frac{1}{9}X_{3,1,0} + \frac{3}{9}X_{3,1,1} + \frac{1}{9}X_{3,2,0} + \frac{3}{9}X_{3,2,1} \end{aligned}$$

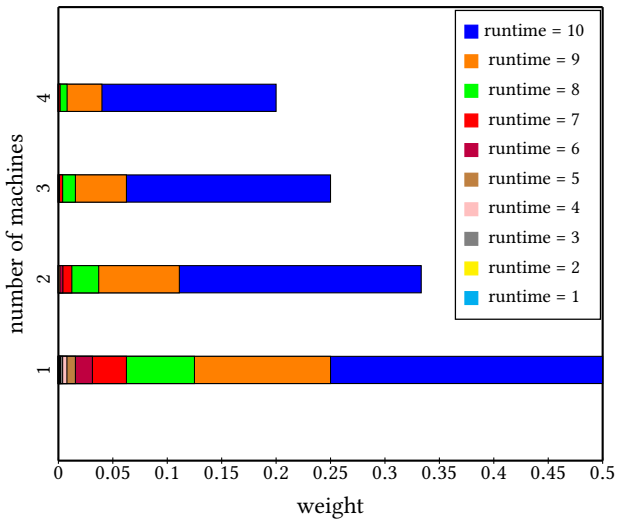


Figure 9: Display of weights for various numbers of machines and end times (Example E_1)

3.3.2 Total solution. Together A, B, C and D form the actual formula, the QUBO-problem, the solution of which is the solution of the actual problem, a valid and optimal distribution of the transactions to the different machines, so that if possible there are no idle times and if possible each transaction has no waiting time:

$$P = A + B + C + D$$

A valid solution always assumes the value $-n$ for $A + B + C$ and is increased by the value of D whenever an optimal solution is reached.

For the example configuration E_2 of Section 3.1.1, the formula P is minimized if all three constraints for a valid schedule are satisfied and the weights across all variables are minimal. In this example, the formula is minimized by four different variable assignments, all of which represent a valid schedule (here, only the variables with the value 1 are specified, the rest take the value 0 accordingly):

$$\begin{aligned} & X_{1,1,0}, X_{2,2,0}, X_{3,2,1} \\ & X_{1,1,0}, X_{2,2,1}, X_{3,2,0} \\ & X_{1,2,0}, X_{2,1,0}, X_{3,1,1} \\ & X_{1,2,0}, X_{2,1,1}, X_{3,1,0} \end{aligned}$$

The solution represents which transaction is started when and on which machine. P assumes the following value for all four different variable assignments:

$$P = A + B + C + D = -3 + 0 + 0 + \frac{7}{9} = -2\frac{2}{9}$$

If the offset is added (optional), then the result is:

$$P = A + B + C + D = -3 + 0 + 0 + \frac{7}{9} + 3 = \frac{7}{9}$$

Both solutions are correct, for the purpose of simplicity the offset is omitted here. The solution is illustrated in Figure 10.

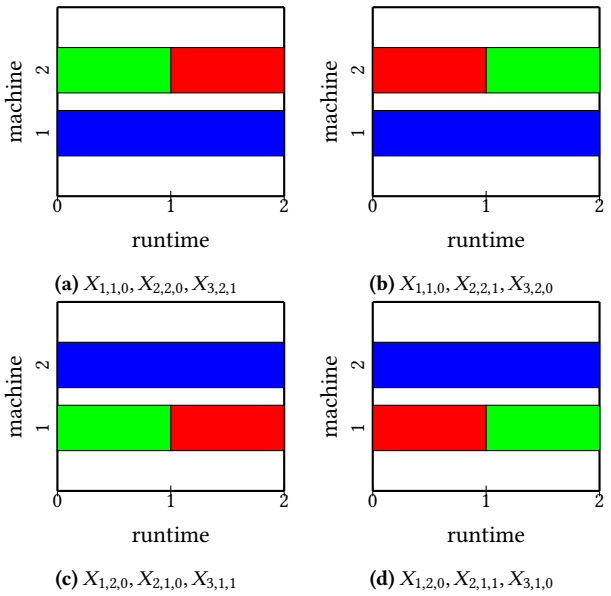
3.3.3 Minimization of execution time. The (maximal) execution time R is a critical parameter during generation of the formula and its solving: If R is too large, then we need too many unnecessary variables, which increases the problem size and hence also the time of preprocessing (but we still retrieve the correct solution). If R is too small, then we do not obtain any solution. In the latter case, we need to rerun the formula generation and solving it on a quantum annealer with an increased R.

The execution time R should therefore be reasonably estimated in advance to avoid unnecessarily many runs on a quantum annealer. Since a quantum annealer executes many runs of the same problem, by slowly increasing the previously estimated execution time, the minimum execution time allowed by a valid schedule can be determined. The lower bound of the execution time is obviously the sum of all lengths divided by the number of machines:

$$R = \frac{\sum_{i=1}^{n} l_i}{k}$$

This value can be estimated even better when considering the length of transactions. Two transactions with lengths 1 and 9 cannot be executed by two machines in time $R = (1+9)/2 = 5$. A more reasonable estimation is:

$$R = \max\{\frac{\sum_{i=1}^{n} l_i}{k}, \max_{i \in \{1,\ldots,n\}} l_i\}$$



(a) $X_{1,1,0}, X_{2,2,0}, X_{3,2,1}$ **(b)** $X_{1,1,0}, X_{2,2,1}, X_{3,2,0}$

(c) $X_{1,2,0}, X_{2,1,0}, X_{3,1,1}$ **(d)** $X_{1,2,0}, X_{2,1,1}, X_{3,1,0}$

Figure 10: Different Schedules (transaction 1 in blue, transaction 2 in green and transaction 3 in red) for example configuration E_2

For the example configuration E_2 of Section 3.1.1:

$$R = \max\{\frac{4}{2}, 2\} = 2$$

If there is one very long transaction and many short ones, the maximum execution time is obviously defined by the length of the very long transaction. By slowly increasing R the minimum execution time and a suitable schedule are found. A valid schedule is identified by its value, since this value does not differ significantly from $-n$, but is significantly larger for an invalid schedule.

4 IMPLEMENTATION OF THE QUBO-PROBLEM

We implement the QUBO-problem described in Section 3.2 by using the Ocean Software of the company D-Wave [2]. The Ocean Software consists of a set of tools that help to formulate a problem for quantum annealers and solves the problem using either a classical computer or a quantum annealer. By only adapting the code for communicating with the quantum annealer, the same code is runnable on a classical computer as well as on a quantum annealer with minimal code modification. The software provides tools for communication with the quantum annealer, for problem formulation through various constraints, for problem solving through various approaches, for embedding the problem on the quantum annealer and many more. Additionally, the software "PyQUBO" [5] is used, which allows to design QUBO formulations from mathematical expressions. Since both software is developed for Python, the implementation is also written in Python. The packages "dimod" [3] (for supporting QUBO models) and "dwave-neal" [4] (for comparing the runtimes of quantum with those of the simulated annealing) are additionally used. The implementation calculates the QUBO-problem from a list of transactions and their lengths, a number of machines and a list of blocking transactions. The execution time R for these transactions is estimated in advance according to Section 3.3.3. The resulting QUBO-problem is precisely solvable by testing all possible variable assignments and thus those with the lowest total value represent the optimal and valid solution. For determining a solution on a classical computer the Simulated-Annealing-Solver [4], one of the Ocean Software tools, is used. For quantum annealing the D-Wave Quantum-Annealer [9] offered as cloud service is used.

5 EVALUATION

In this section, we analyze the complexity (see Section 5.1) in terms of preprocessing time (see Section 5.1.1) and required qubits (see Section 5.1.2), and experimental results comparing simulated with quantum annealing (see Section 5.2).

5.1 Formal analysis

A time analysis for quantum computers as a function of problem variables such as number of transactions or number of machines is generally not possible, since for quantum computers the number of annealing runs as well as the times per annealing run and the times for readout can be determined in advance.

We first determine the number of required qubits and the time of preprocessing in terms of a complexity analysis in Section 5.1.1. The problem sizes remain n for the number of transactions, k for the number of machines and R for the maximum execution time.

5.1.1 Preprocessing time. $O(n \cdot k \cdot R)$ binary variables are used within the formula for distributing transactions to k machines to be minimized. A and D contain $O(n \cdot k \cdot R)$ terms, B contains $O(n^2 \cdot k \cdot R^2)$ terms and C in case all transactions block each other $O((n \cdot k \cdot R)^2)$ terms. Calculating the weights for D is performed in constant time. For the worst case, we overall achieve a quadratic time for preprocessing depending on the problem sizes:

$$O((n \cdot k \cdot R)^2)$$

For cases, where the number of conflicting transactions is at most linear to n (which might be the typical case for transaction workloads in operational systems), the number of terms is dominated by the number of terms in B (assuming $n > k$) and is hence $O(n^2 \cdot k \cdot R^2)$.

5.1.2 Required qubits. In quantum annealers from D-Wave, qubits are arranged in the Chimera structure, which simply states that each qubit is entangled with a maximum of six other qubits. If the binary variables occur in quadratic terms, the entanglements are weighted accordingly. Since only entanglements can be weighted, but no single qubits and binary variables usually occur weighted with more than six other variables together, several qubits are defined as one variable. The individual weights are now distributed over the entanglements of the qubits of a variable. For the entanglements one qubit per variable is selected and the entanglement is weighted. Consequently, the number of qubits required depends on how many connections the binary variables have with each other. A and D do not contain quadratic terms and are therefore not important. In B each variable is connected to a maximum of $O(n \cdot R)$ others, in C

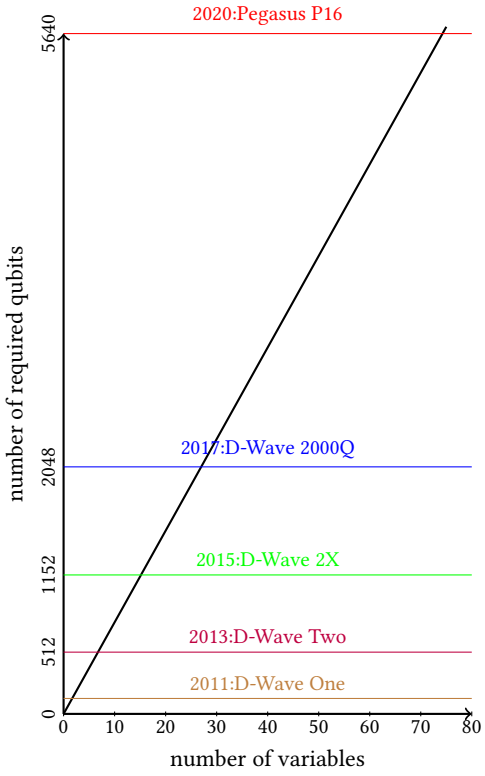


Figure 11: Required qubits depending on the number of variables and available qubits of D-Wave Quantum-Annealers

to a maximum of $O(n \cdot k \cdot R)$. The number of required qubits thus increases quadratically dependent on the problem sizes:

$$O((n \cdot k \cdot R)^2)$$

The maximum number of variables is thus limited by the root of the number of qubits. Figure 11 shows the number of required qubits as a function of the number of variables, where the horizontal lines represent the numbers of qubits of available and future quantum annealers.

5.2 Experimental analysis

For the experimental analysis, the time of solving is measured using simulated annealing and quantum annealing. We present the details about the test configurations (i.e., number of cores, maximum execution times, transactions and conflicts between them, their start times and lengths) in the Appendix.

For quantum annealing we use the cloud-service available at [9], which offers free access to a D-Wave 2000Q quantum computer for some time. The offered free time was enough to run the experimental evaluation described in Section 5.2.1.

The experiments for preprocessing and simulated annealing are run on an i7-4510U dual core CPU with 2.0 GHz and 8 GB main memory running Windows 10. Every simulated annealing run and quantum annealing run was measured a hundred times and the lowest value is determined as an (almost) optimal solution. We

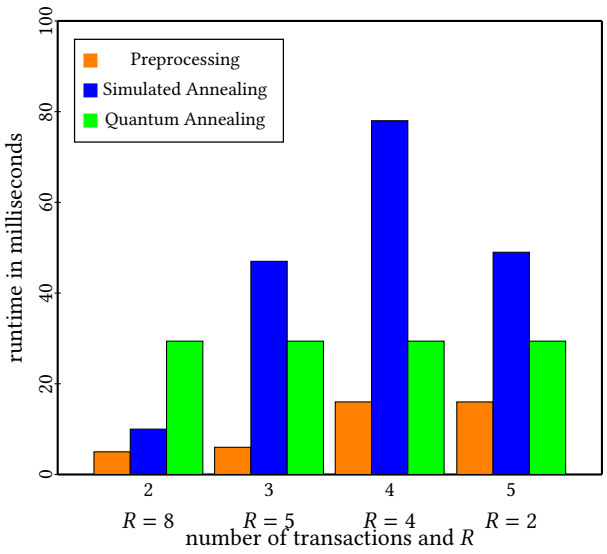


Figure 12: Runtimes of simulated annealing versus quantum annealing for $k = 2$

present the overall total time of these hundred runs in the following sections.

5.2.1 Runtimes Quantum Annealing. Due to the limited number of qubits only small problem instances could be measured by quantum annealing (see Figure 12). The preprocessing time for generating the formula is the same for simulated as well as for quantum annealing, as both take the formula as input and find a minimal solution for it.

For every quantum annealing run the determined runtime was 20 microseconds as well as 274 microseconds for the readout. Measuring a hundred times will then take 29.4 milliseconds. While for the runtime of two transactions simulated annealing is still faster than quantum annealing, already problem sizes with 3 transactions and above are faster with quantum annealing with an observed speedup up to approx. 2.65. Once quantum annealers supporting more qubits are available, larger problem sizes can be solved with quantum annealing. As the times for quantum annealing and readout are constant, but simulated annealing runs become slower for larger problem sizes due to the more complex evaluation of the formula, larger speedups will be probably achieved when running on future quantum annealers.

5.2.2 Runtimes Simulated Annealing. For simulated annealing, we don't have limitations concerning the number of variables used in the formula to minimize. Hence we measure the runtimes of simulated annealing for larger problem sizes (see Figures 13 and 14). With a total of around 30 milliseconds the (constant) runtime of pure optimization by the quantum annealer is significantly better than that of simulated annealing even for relatively small problem instances. Assuming that also future quantum annealers will have approximately the same (constant) runtimes, speedups of quantum in comparison to simulated annealing up to 3300 for the problem sizes in Figures 13 and 14 are achievable.

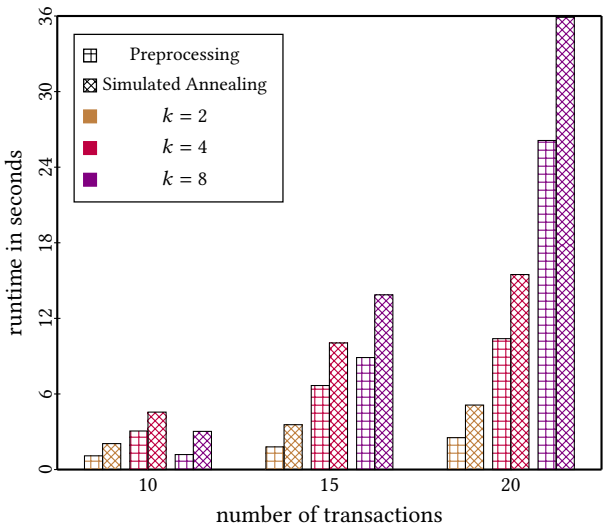


Figure 13: Runtimes for $R = 20$

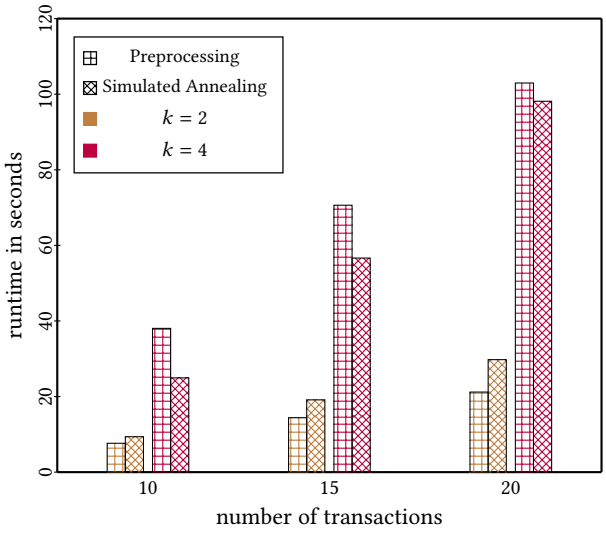


Figure 14: Runtimes for $R = 50$

6 SUMMARY AND CONCLUSIONS

In this paper, we show how to transform the problem of distributing transactions to arbitrary machines into an optimization problem that can be solved by quantum computers. In more detail, we transform a corresponding problem instance into a QUBO-problem to be run on a quantum annealer that is a special type of quantum computer for solving QUBO-problems. In the evaluation, the number of qubits required for the QUBO-problem of distributing transactions and the time needed for the transformation are analyzed. Furthermore, we conduct an experimental analysis comparing the runtimes of a quantum annealer with the one of simulated annealing on a classical computer. We already achieved speedups of up to over 2.6. However, the number of qubits of current quantum annealers has so far limited the number of possible problem sizes to rather small ones, which promises higher speedups for future generations of quantum annealers supporting many more qubits. Thus a real use of quantum computers for optimization problems of this kind is likely to continue in the future.

In our future work we will investigate how to improve e.g. the preprocessing time by indexing subformulas in some kind of cache for already determined transaction setups, such that formulas can be built based on previous ones by just iterating through the pairs of blocking transactions. Furthermore, we may also consider to accelerate other transaction models and synchronization problems as well as other areas of database research by quantum computers.

REFERENCES

[1] 2017. The D-Wave 2000Q™ System. https://www.dwavesys.com/d-wave-two-system
[2] 2017. D-Wave's Ocean Software. https://ocean.dwavesys.com
[3] 2017. dimod. https://github.com/dwavesystems/dimod
[4] 2017. dwave-neal. https://github.com/dwavesystems/dwave-neal
[5] 2017. PyQUBO. https://github.com/recruit-communications/pyqubo
[6] Adriano Barenco, David Deutsch, Artur Ekert, and Richard Jozsa. 1995. Conditional quantum dynamics and logic gates. *Physical Review Letters* 74, 20 (1995), 4083.
[7] Carlos Barrón-Romero. 2015. Classical and Quantum Algorithms for the Boolean Satisfiability Problem. *arXiv preprint arXiv:1510.02682* (2015).
[8] Thomas M Connolly and Carolyn E Begg. 2005. *Database systems: a practical approach to design, implementation, and management.* Pearson Education.
[9] D-Wave Systems. 2020. Take the Leap. https://www.dwavesys.com/take-leap
[10] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. 2000. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106* (2000).
[11] M. R. Garey, D. S. Johnson, and Ravi Sethi. 1976. The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research* 1, 2 (1976), 117–129.
[12] Fred Glover, Gary Kochenberger, and Yu Du. 2019. Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models. *4OR* 17, 4 (2019), 335–371.
[13] Sven Groppe. 2020. Emergent models, frameworks, and hardware technologies for Big data analytics. *The Journal of Supercomputing* 76, 3 (2020), 1800–1827. https://doi.org/10.1007/s11227-018-2277-x
[14] Sven Groppe and Jinghua Groppe. 2020. Hybrid Multi-Model Multi-Platform (HM3P) Databases. In *Proceedings of the 9th International Conference on Data Science, Technology and Applications (DATA).*
[15] Lov K Grover. 1996. A fast quantum mechanical algorithm for database search. *arXiv preprint quant-ph/9605043* (1996).
[16] Lov K Grover. 1997. Quantum computers can search arbitrarily large databases by a single query. *Physical review letters* 79, 23 (1997), 4709.
[17] Tad Hogg. 2003. Adiabatic quantum computing for random satisfiability problems. *Physical Review A* 67, 2 (2003), 022314.
[18] Anastasia Marchenkova. 2016. *What's the difference between quantum annealing and universal gate quantum computers?* https://medium.com/quantum-bits
[19] Martin Pincus. 1970. Letter to the Editor-A Monte Carlo Method for the Approximate Solution of Certain Types of Constrained Optimization Problems. *Operations Research* 18, 6 (1970), 1225–1228.
[20] Sudip Roy, Lucja Kot, and Christoph Koch. 2013. Quantum databases. In *Proc. CIDR.*
[21] Peter W Shor. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science.* Ieee, 124–134.
[22] Tobias Stollenwerk and Achim Basermann. 2016. Experiences with scheduling problems on adiabatic quantum computers. In *Proceedings of the 1st International Workshop on Post-Moore Era Supercomputing (PMES), Future Technologies Group Technical report FTGTR-2016-11.* 45–46.
[23] A. Thomson, T. Diamond, S.-C. Weng, K. Ren, P. Shao, and D. J. Abadi. 2012. Calvin: Fast Distributed Transactions for Partitioned Database Systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)* (Scottsdale, Arizona, USA) *(SIGMOD '12).* 1–12. https://doi.org/10.1145/2213836.2213838
[24] Immanuel Trummer and Christoph Koch. 2016. Multiple Query Optimization on the D-Wave 2X Adiabatic Quantum Computer. *Proc. VLDB Endow.* 9, 9 (May 2016), 648–659. https://doi.org/10.14778/2947618.2947621
[25] Davide Venturelli, Dominic J. J. Marchand, and Galo Rojo. 2015. Quantum Annealing Implementation of Job-Shop Scheduling. *arXiv* (2015). arXiv:1506.08479 [quant-ph] https://arxiv.org/abs/1506.08479

APPENDIX

We present the test configurations for figures 12,13 and 14 in the following tables. For each test configuration, we assume $T = \{t_1, ..., t_n\}$ and $M = \{m_1, ..., m_k\}$. The last column of the following table contains the number of required variables.

Fig.	k	n	R	O	$l_1, ..., l_n$	$r_1, ..., r_n$	**req. var.**
12	2	2	8	$\{\}$	8, 4	0, 4	8
		3	5	$\{(t_1, t_3)\}$	4, 5, 1	1, 0, 4	10
		4	4	$\{(t_2, t_4)\}$	3, 2, 1, 2	1, 2, 3, 2	16
		5	2	$\{(t_1, t_2), (t_4, t_5)\}$	1, 1, 1, 1, 1	1, 1, 1, 1, 1	10
13	2	10	20	$\{(t_1, t_3), (t_3, t_4), (t_7, t_8)\}$	5, 7, 1, 1, 3, 4, 4, 2, 8, 5	15, 13, 19, 19, 15, 16, 16, 18, 12, 15	320
	4			$\{(t_1, t_2), (t_9, t_{10})\}$	10, 14, 8, 5, 4, 7, 6, 10, 3, 13	10, 6, 12, 15, 16, 13, 14, 10, 17, 7	480
	8			$\{(t_1, t_2), (t_3, t_4), (t_5, t_6)\}$	16, 20, 5, 15, 4, 6, 14, 10, 6, 12	4, 0, 15, 5, 16, 14, 6, 10, 14, 8	736
	2	15		$\{(t_2, t_3), (t_4, t_6), (t_5, t_{10}), (t_1, t_{15})\}$	4, 2, 3, 1, 4, 6, 2, 1, 5, 1, 1, 3, 2, 4, 1	16, 18, 17, 19, 16, 14, 18, 19, 15, 19, 19, 17, 18, 16, 19	520
	4			$\{(t_1, t_2), (t_3, t_4), (t_7, t_8)\}$	6, 4, 3, 1, 2, 8, 7, 10, 6, 11, 1, 3, 6, 4, 8	14, 16, 17, 19, 18, 12, 13, 10, 14, 9, 19, 17, 14, 16, 12	880
	8			$\{(t_2, t_4), (t_3, t_6), (t_3, t_{14})\}$	9, 12, 5, 3, 8, 14, 3, 7, 8, 1, 2, 4, 20, 1, 3	11, 8, 15, 17, 12, 6, 17, 13, 12, 19, 18, 16, 0, 19, 17	1600
	2	20		$\{(t_5, t_{12}), (t_6, t_{18}), (t_7, t_{20}), (t_8, t_{20})\}$	1, 1, 1, 2, 4, 1, 1, 2, 1, 1, 2, 2, 2, 1, 1, 1, 1, 6, 5, 4	19, 19, 19, 18, 16, 19, 19, 18, 19, 19, 18, 18, 18, 19, 19, 19, 19, 14, 15, 16	720
	4			$\{(t_1, t_2), (t_2, t_{17}), (t_5, t_9)\}$	3, 4, 1, 2, 4, 1, 1, 7, 1, 10, 1, 9, 5, 3, 8, 5, 2, 7, 1, 8	17, 16, 19, 18, 16, 19, 19, 13, 19, 10, 19, 11, 15, 17, 12, 15, 18, 13, 19, 12	1268
	8			$\{(t_2, t_4), (t_8, t_{19}), (t_{11}, t_{13})\}$	7, 16, 8, 4, 11, 4, 6, 5, 7, 12, 19, 13, 1, 9, 1, 2, 3, 4, 15, 14	13, 4, 12, 16, 9, 16, 14, 15, 13, 8, 1, 7, 19, 11, 19, 18, 17, 16, 5, 6	1912
14	2	10	50	$\{(t_2, t_{10}), (t_3, t_4)\}$	15, 17, 10, 10, 4, 9, 8, 12, 2, 13	35, 33, 40, 40, 46, 41, 42, 38, 48, 37	800
	4			$\{(t_1, t_2), (t_9, t_{10})\}$	20, 21, 12, 25, 24, 16, 27, 15, 18, 22	30, 29, 38, 25, 26, 34, 23, 35, 32, 28	1200
	2	15		$\{(t_1, t_2), (t_3, t_{11}), (t_5, t_{10})\}$	6, 7, 3, 1, 12, 8, 7, 10, 4, 11, 1, 13, 6, 4, 8	44, 43, 47, 49, 38, 42, 43, 40, 46, 39, 49, 37, 44, 46, 42	1298
	4			$\{(t_1, t_3), (t_3, t_4), (t_7, t_9), (t_{11}, t_{12})\}$	15, 12, 9, 1, 12, 22, 23, 14, 9, 8, 14, 17, 24, 13, 7	35, 38, 41, 49, 38, 28, 27, 36, 41, 42, 36, 33, 26, 37, 43	2200
	2	20		$\{(t_4, t_5), (t_5, t_6), (t_{12}, t_{13}), (t_{17}, t_{18})\}$	3, 4, 1, 2, 4, 1, 10, 7, 1, 11, 1, 9, 8, 3, 8, 5, 2, 7, 1, 12	47, 46, 49, 48, 46, 49, 40, 43, 49, 39, 49, 41, 42, 48, 42, 45, 48, 43, 49, 38	1800
	4			$\{(t_3, t_{17}), (t_5, t_7), (t_6, t_8)\}$	12, 17, 19, 4, 11, 22, 3, 14, 23, 2, 9, 12, 2, 12, 12, 4, 5, 2, 3, 14	38, 33, 31, 46, 39, 28, 47, 36, 27, 48, 41, 38, 48, 38, 38, 46, 45, 48, 47, 36	3192

Towards A Universal Approach for Semantic Interpretation of Spreadsheets Data

Nikita O. Dorodnykh*
Aleksandr Yu. Yurin
tualatin32@mail.ru
iskander@icc.ru
Matrosov Institute for System Dynamics and Control Theory,
Siberian Branch of the Russian Academy of Sciences
Irkutsk, Russia

ABSTRACT

Spreadsheets are a popular way to represent and structure data and knowledge; in this connection semantic interpretation of spreadsheets data has become an active area of scientific research. In this paper, we propose a new approach for semantic interpretation of data extracted from spreadsheets with arbitrary layouts and styles. Analyzed spreadsheets are presented in the MS Excel format. In particular, our approach includes two stages: analyzing and transforming source spreadsheets to spreadsheets in a relational canonicalized form; annotating canonical spreadsheets by entities from a knowledge graph. At the first stage we use a rule-based approach implemented in the form of a domain-specific language called Cells Rule Language (CRL), and an original form of a canonical table. At the second stage we use an aggregated method for defining similarity between candidate entities and cell values that consists of the sequential application of five metrics and combining ranks obtained by each metric. Algorithms of each stage are implemented in the form of special software: TabbyXL and TabbyLD respectively. DBpedia is used as a knowledge graph. Experimental evaluations of our proposals are obtained for T2Dv2 and Troy200 corpuses, and they demonstrates the applicability of our approach and software for semantic spreadsheet data interpretation. The feature of the approach is its universality due to the use of the language for describing spreadsheets transformation rules, as well as an original canonical form. This feature provides processing large volumes of heterogeneous spreadsheets in various domains. This work is a part of the Tabby research project for software development of recognition, extraction, transformation and interpretation of data from spreadsheet tables with arbitrary layouts and styles.

CCS CONCEPTS

• **Information systems → Information retrieval**; **Information extraction**; **Semantic web description languages**.

*Both authors contributed equally to this research.

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410609

KEYWORDS

semantic interpretation, named entity recognition, named entity linking, spreadsheet data, linked data, knowledge graph, dbpedia

ACM Reference Format:
Nikita O. Dorodnykh and Aleksandr Yu. Yurin. 2020. Towards A Universal Approach for Semantic Interpretation of Spreadsheets Data. In *24th International Database Engineering Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3410566.3410609

1 INTRODUCTION

Spreadsheets suggest a convenient and visual form for structured representation of information. For this reason they are widely distributed, and their number, only in the Internet, is about 150 million [1]. Spreadsheets contain useful knowledge in many domains (for example, business, analytics, engineering, data-driven research, and others), and spreadsheets data can be effectively used for software, knowledge bases, conceptual models and ontologies engineering.

Web tables and spreadsheets can be used as the main artifacts of data analysis. However, the lack of understanding the semantic structure and meaning of their content may reduce the effectiveness of this process. Thus, the restoration of this semantic interpretation is a relevant task for data cleaning and integration, data mining and knowledge discovery domains. The main aims of semantic spreadsheet interpretation are to provide the automated intelligent processing the data [2] and increase the general semantic interoperability of spreadsheets in the context of e-business [3].

In general, semantic spreadsheets data interpretation is a comparison of tabular data with a knowledge graph (e.g., Wikidata, DBpedia or YAGO) and selecting semantic tags for source spreadsheet elements (Figure 1). The complexity of this task is results from incompleteness, ambiguity, or lack of metadata (e.g., table heading names). In turn spreadsheets may also have arbitrary layouts and styles, and describe both one class (an entity) and many instances (objects) belonging to one or more classes. So, to properly process and annotate these spreadsheets the specialized algorithms and software oriented to a certain class (or a subset) of spreadsheets layouts should be designed.

In this paper, we propose a two-stage approach for semantic interpretation of data extracted from spreadsheets in MS Excel format with arbitrary layouts. The proposed approach is implemented in the form of the following pipeline: TabbyXL [4] is used to extract relational data in a canonicalized form from spreadsheets with

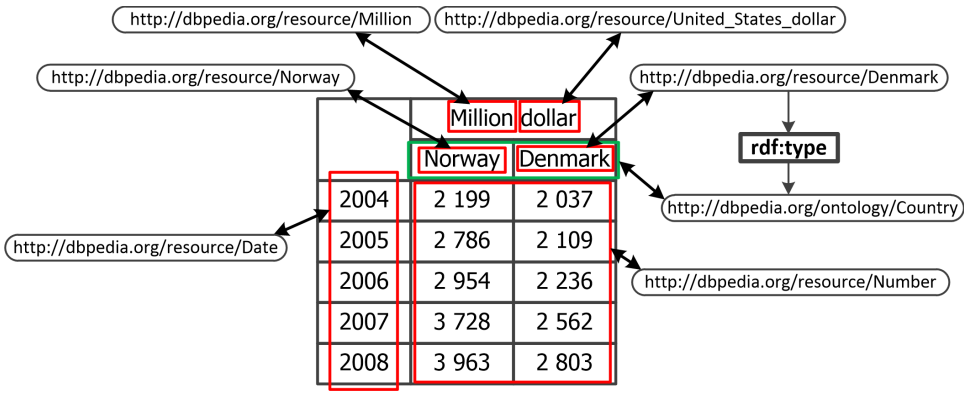


Figure 1: An illustrative example of semantic interpretation of spreadsheet data.

an arbitrary layout and styles; TabbyLD [5] is used to annotate both a single spreadsheet and a set of spreadsheets presented in a canonicalized form. We also provide an experimental evaluation that generally shows the overall applicability of our approach.

The originality of this approach lies in the fact that we offer a solution for analyzing and interpreting data from arbitrary tables that does not focus on any specific domain or a specific spreadsheet dataset (a corpus). This solution gives flexibility in analyzing and processing various spreadsheets, and also provides convenient tools for non-programmers to obtain a visual result.

The paper is organized as follows. Section 2 provides an analytical overview of related works and defines our contributions. Section 3 describes the proposed two-stage approach for semantic spreadsheet data interpretation including a brief description of implementation. Section 4 illustrates the experimental evaluation of our approach and discussion of results, while Section 5 presents concluding remarks.

2 BACKGROUND

2.1 Table data analysis and processing

Table data analysis and processing are a challenge which includes the following main tasks:

- *Table extraction* is retrieving tables from various information sources (e.g., documents, web pages, etc.), and their subsequent storing in a certain format. This task usually contains such steps as table detection, table recognition (segmentation), and functional (role) and structural table analysis. The result of this step is a set of extracted tables.
- *Table interpretation* is restoration of table semantics, i.e. linking its content with the target concepts from a knowledge graph (a global taxonomy) or some domain ontology.

There are four main subtasks of semantic table interpretation:

- *Named entity and literal columns detection.*
- *Entity linking* (recognizing named entities extracted from table cells and linking them with entities from a knowledge graph).
- *Column type identification* (defining a relevant type (class) from a knowledge graph that characterizing data in the table column based on the entire set of cell values of this column).
- *Relation extraction* (defining a relationship between two columns of a source table in the form of a reference property from a knowledge graph).

Table extraction and interpretation results are a set of tabular data marked with labels (tags) from a target knowledge graph. These tables are suitable for further machine processing. In particular, the following tasks can be solved:

- *Table retrieval* is a task of using a ranked list of tables for response to a user request. In this case web tables mainly are used, and a user request can be represented as a sequence of keywords or a table itself.
- *Knowledge base augmentation* is a task of exploring, constructing, and augmenting new knowledge bases or some knowledge base elements (e.g., a set of specific facts or a set of templates), as well as expanding existing knowledge bases with the use of tabular data.

2.2 Related works

Many approaches and tools aimed the semantic interpretation of spreadsheets data have been proposed in recent years. These approaches are intended for linking tabular content to external entities contained in a knowledge graph. Typically, they focus on analyzing the natural language content of spreadsheets and it context.

In particular, a new technique for table interpretation based on a scalable graphical model using entity similarities is proposed in [6]. This model uses label similarities as priors, and then updates its likelihood scoring to maximise the coherence of entity assignments across the rows using loopy belief propagation. This technique focuses on web tables containing named entities, i.e. tables in which one column contains named entities (a key column), and all the rest contain properties. Candidate entities are sampled for each key column value, and label matches are ranked using a method of length-normalised smoothed (TF-IDF). The similarity of labels with entities from a set of candidate entities is defined using a row proximity (Jacquard coefficient), cover and salience definitions.

Efthymiou [7] presents a hybrid approach for a semantic interpretation of tabular data consisting of a combination of three methods: (1) a lookup-based method which relies on the minimal entity context provided in web tables to discover correspondences to a

knowledge graph; (2) a semantic embeddings method that exploits a vector representation of the rich entity context in a knowledge graph to identify the most relevant subset of entities in a web table. This method is based on the global disambiguation technique used in the DoSeR framework, where similarity between entities is computed as the cosine distance between their vector representations. Authors use the PageRank algorithm in addition to this method; (3) an ontology matching method which exploits schematic and instance information of entities available both in a knowledge graph and a web table.

In [8] eight basic tasks related to the normalization and interpretation of web tables from the WDC Web Table Corpus 2015 (WTC) are described. In particular, the interpretation method uses an entity linking algorithm for associating values in table cells with entities from DBpedia (binding using URI), as well as a literal linking algorithm.

Zhang [9] proposes a method and TableMiner+ software for semantic interpretation of web tables. This method improves the accuracy of annotations by using contextual information both inside a source table (considering columns, cells and rows of a source table) and outside ones (considering a web page name, a table name, paragraphs and unstructured text, semantic markup, etc.). Zhang method also reduces computational complexity using an incremental bootstrapping approach for forming a set of candidate entities based on a selection of table cells with less ambiguity (less ambiguous cells).

A new approach for annotating web tables based on an entity linking procedure is proposed in [10]. Authors use different knowledge graphs and concepts with a "sameAs" relationship. Two main metrics are used to define similarity between string values and entities from a set of candidates: a string similarity and mention-entity context similarity. The usual finding of a relationship between entities and an entity-entity context similarity metric can be used to determine the relationship between candidate entities.

Other examples of solving the task of semantic table data interpretation are presented in [11–21].

Despite the existence of examples of methods and software for semantic interpretation of table data there is no any universal approach independent of spreadsheet types and layouts. As a rule, all considered approaches process relational tables with a certain structure. Such tables should describe data belonging to a single class (category), where each column is associated with a property from a knowledge graph, and all rows of a table are objects (instances) of this class. It should also be noted that mainly the web tables are analyzed; in turn spreadsheets in the CSV or Excel formats are considered in short, although the interest to this topic is rising. In particular, a semantic web challenge on tabular data to knowledge graph matching was held at the 19th International Semantic Web Conference (ISWC) [22], which confirms the relevance of the task considered in this paper.

Thus, in our paper we make an attempt to build a universal approach for semantic interpretation of data extracted from spreadsheets with an arbitrary layout presented in the Excel format. Our contributions are the following:

- A new two-stage approach for semantic data interpretation of spreadsheets with arbitrary layouts. First, spreadsheets are analyzed and converted to a relational canonicalized form. Then we apply five heuristic-based methods to determine the similarity between elements (labels) from a source canonical spreadsheet and entities (concepts) from a target knowledge graph. Using this approach we can process large volumes of heterogeneous spreadsheets in various domains.
- An adapted version of the standard concept of named entity recognition from the information extraction domain in the context of analysis of canonical spreadsheets; this adapted version uses the recognition results to restore a semantic data understanding by means of a named entity linking. Using this concept we more efficiently process spreadsheets containing both specific literal values (e.g., data of any dimensions) and named entities.
- A complete pipeline for analyzing, transforming, and interpreting data extracted from arbitrary spreadsheets, consisting of the TabbyXL [4] tool and prototype of web-based software called the TabbyLD [5] that implements our two-step approach. TabbyXL is used for the first time in the task of semantic interpretation of spreadsheet data and provides an intermediate link to obtain spreadsheets in a canonicalized form. TabbyLD has an intuitive graphic user interface and designed for a wide range of users including non-programmers.

Advantages of the proposed universal approach are the following:

- An ability to analyze various types of spreadsheets with any arbitrary layout and styles.
- A support for processing all spreadsheet content, including all column cells, rather than cells for a specific column.
- An ability to interpret data in cells of various types, which are both literal values and named entities.
- A support for processing MS Excel and CSV spreadsheets and the potential use of various knowledge graphs for the process of semantic interpretation of spreadsheet data.

3 PROPOSED APPROACH

Our approach for semantic spreadsheet data interpretation consists of two main stages:

Stage 1: Obtaining canonical spreadsheets. Obtaining spreadsheets in a relational canonicalized form based on the analysis and transformation of spreadsheets with arbitrary layouts.

Stage 2: Annotating canonical spreadsheets. Annotating canonical spreadsheets using a knowledge graph. This stage contains the following steps:

- *Data cleaning* includes preparing a canonical spreadsheet for the annotation by formatting each value in cells.
- *Named Entity Recognition* includes identifying named entities in cells of a canonical spreadsheet and assigning each spreadsheet cell of corresponding type.
- *Named Entity Linking* is matching each cell values of a canonical spreadsheet with entities from a target knowledge graph (i.e., DBpedia).
- *Generating annotated spreadsheets* with links (tags) of concepts from a target knowledge graph (DBpedia).

Our approach can be illustrated in the form of a diagram shown in Figure 2.

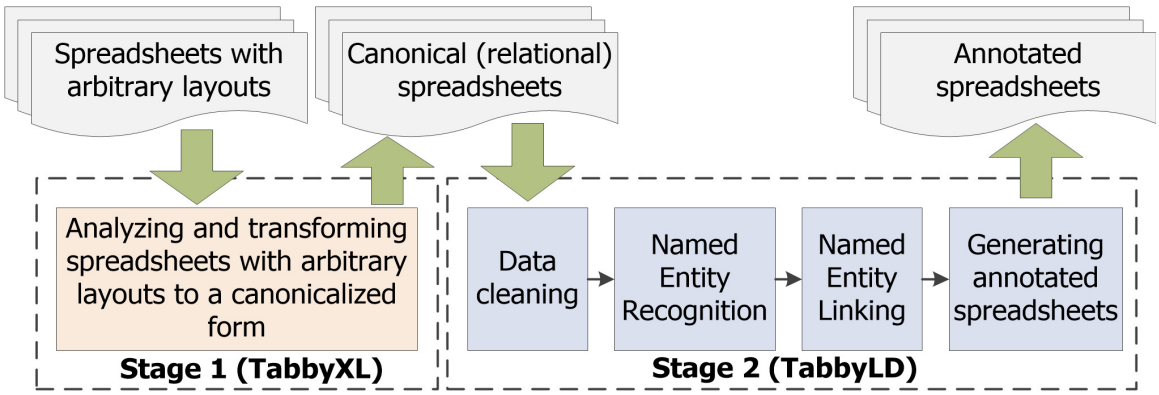


Figure 2: A two-stage approach for semantic interpretation of spreadsheets data.

Next, let's consider each stage in detail.

3.1 Obtaining canonical spreadsheets

There are many variants for the physical and logical representation of spreadsheets. Spreadsheets can have different layouts, design styles, data representation logic, and others. Such spreadsheets we called arbitrary. In this paper, we propose to use a rule-based approach [23] for the analysis of such spreadsheets and their transformation to a relational canonicalized form. Relational spreadsheets contain high-quality relational data [24] in the form of a set of entities, which could exist in rows (horizontal) or columns (vertical), the remainder of cells contain their descriptive attributes.

Let's formally define the structure of relational spreadsheets represented in a canonicalized form:

$$CT = \left\{D, R^H, C^H\right\}, \tag{1}$$

where D is a data block that describes literal data values (entries) belonging to the same data type (e.g. numeric, date, time or text); R^H is a set of row labels of the category; C^H is a set of column labels of the category. The values in cells for heading blocks can be separated by the " | " symbol to divide categories into subcategories. Thus, the canonical spreadsheet denotes hierarchical relationships between categories (headings).

The rule-based approach uses a table object model, which is designed for representing both a physical structure (a set of cells) and logical data items (three sets of entries (values), labels (keys), and categories (concepts)) of an arbitrary spreadsheet in the process of its analysis. We use a domain-specific language called Cells Rule Language (CRL) for expressing this model and transforming source spreadsheets. The rules expressed in this language are intended to map explicit features (layout, style, and text of cells) of an arbitrary spreadsheet into its implicit semantics (entries, labels, and categories). A set of the rules can be implemented for a specific task characterized by requirements for source and target data. A more detailed description of this approach is presented in [23].

3.2 Annotating canonical spreadsheets

Step 1: Data cleaning. The purpose of this step is to prepare (preprocess) the canonical spreadsheet data for further qualitative entity recognition and linking, in particular, to generate correct SPARQL queries for a target knowledge graph. As a result, cell values are cleared from various "garbage" characters, except for letters and numbers. Multiple spaces are also removed, and single spaces are replaced by the underscore. All mentions in cell are lowercase, with the first letter in the first word being capitalized. We are also trying to decipher existing acronyms.

Step 2: Named Entity Recognition (NER). The procedure for extracting and recognizing named entities contained in canonical spreadsheet cells is carried out in this step. We use the library for natural language processing called Stanford CoreNLP [25] and, in particular, Java implementation of Stanford Named-Entity Recognizer [26]. Stanford NER marks words in the text that are object names such as peoples, companies, cities or countries. Stanford NER defines many classes of named entities. In our work, we use 8 basic types (classes): Location, Country, City, Person, Organization, Number, Percent, Date, and None that is an undefined class. These classes are assigned to each cell in a source canonical spreadsheet, so ones characterize the data that it contains. We divide such specific typed cells into two groups: cells with named entities and cells with literal values. Correspondences between NER labels and cell types are presented in Table 1. It should be noted that cells marked "None" are not recognized by Stanford NER; we define such cells as cells with named entities.

Step 3: Named Entity Linking (NEL). The procedure for linking recognized named entities contained in canonical spreadsheet cells is carried out in this step.

In recent years, a NEL challenge has attracted a great attention from the academic community. It is related to the fact that there are many open large knowledge bases and computing resources that make a NEL task feasible. A NEL aim is to define the identity of entities mentioned in spreadsheet cells with respect to some target knowledge graph. We selected DBpedia [27] as a target knowledge graph. A NEL task involves the formation of a set of candidate entities from DBpedia for each cell value in a source canonical spreadsheet (candidate search phase) and disambiguation.

At the candidate search phase, we first try to find an exact match for each cell value with an entity from DBpedia. Then we form a set of candidate entities (E) by compiling SPARQL queries against

Table 1: The correspondences between Stanford NER labels and DBpedia entities

NER-label	Cell type	DBpedia entity (URI)
None	named-entity	–
Location	named-entity	http://dbpedia.org/ontology/Location
Country	named-entity	http://dbpedia.org/ontology/Country
City	named-entity	http://dbpedia.org/ontology/City
Person	named-entity	http://dbpedia.org/ontology/Person
Organization	named-entity	http://dbpedia.org/ontology/Organisation
Number	literal	http://dbpedia.org/ontology/Number
Percent	literal	http://dbpedia.org/ontology/Percent
Date	literal	http://dbpedia.org/ontology/Date

DBpedia for matching a string to a pattern. At the same time, we select only the first 100 candidates for performance reasons.

At the candidate search phase, we first try to find an exact match for each cell value with an entity from DBpedia. Then we form a set of candidate entities (E) by compiling SPARQL queries against DBpedia for matching a string to a pattern. At the same time, we select only the first 100 candidates for performance reasons.

Since a cell value can refer to several entities from a set of candidates, it is rather difficult to choose a suitable reference entity from this set for linking. We propose an aggregated method for defining similarity between candidate entities and cell value to disambiguation. This method consists of the sequential application of five metrics and combining ranks obtained by each metric. Next, let's consider each metric in more detail.

1) String similarity. This metric is used for selecting the most reference entity e_i from a set of candidates E for the cell value v_j based on the maximum similarity of sequence of characters in the accordance of the Levenshtein distance:

$$\psi_1(e_i) = LevenshteinDistance\left(v_j, e_i\right), v_j \in CT, e_i \in E, \quad (2)$$

where $\psi_1(e_i)$ is a function for calculating the Levenshtein distance; v_j is a string value of j-cell in a source canonical spreadsheet; e_i is a string name of i-entity from a set of candidates E.

2) NER label similarity. This metric is based on information about already recognized named entities and cell types form the Step 2. All assigned NER labels are mapped to entities (classes and instances) from DBpedia (see Table 1). Further, if a cell has a literal type, then we linking a value v_j in this a j-cell to a specific entity defined in Table 1.

If the current cell has a named-entity type, then we suggest to find relationships between each entity e_i from a set of candidates E and a specific class defined in Table 1. Moreover, these relationships can be transitive. For example, the *"Hong_Kong_University_Press"* entity is only an instance for the *"Publisher"* class and has no a direct relationship with the *"Organization"* class, but this relationship can be restored through the transitive dependency of subclasses using the *"rdfs:subClassOf"* construction: *"Publisher"* → *"Company"* → *"Organization"*.

If the depth of classes nesting (a distance from this target class relative to a current candidate entity) is defined, then the rank is defined as follows:

$$\psi_2(e_i) = \frac{1}{distanceLevel(c_j)}, e_i \in E, c_j \in C, \quad (3)$$

where $\psi_2(e_i)$ is a function for calculating similarity to classes defined by NER labels for an entity e_i from a set of candidates E; c_j is a name of j-class from a set of classes C found by NER labels (see Table 1); $distanceLevel(c_j)$ is a class c_j distance (depth) in a transitive relationship with entity e_i. Moreover, if candidate entity e_i is an instance of the required class directly, then this distance is 1.

String similarity and NER label similarity are our basic metrics and they are applied to all cells of a source canonical spreadsheet, both for a data block (D) and for heading blocks (R^H and C^H). The following 3 metrics are used only for cell values located in a data block (D). Therefore, we calculate the aggregate rank for candidate entities linked to a heading blocks (R^H and C^H):

We define a reference entity e_i from a set of candidates E based on a combination of ranks obtained by $\psi_1(e_i)$ and $\psi_2(e_i)$ functions:

$$f^H_{agg} = w_1 \times \left(1 - \frac{\psi_1(e_i)}{100}\right) + w_2 \times \psi_2(e_i), \quad (4)$$

where f^H_{agg} is an aggregate rating function for a candidate entity e_i; w_1 is a weighting factor that balances the importance of a rank obtained on the basis of the Levenstein distance calculation; w_2 is a weighting factor that balances the importance of rank obtained on the basis of the NER label similarity. By default, $w_1 = 1$ and $w_2 = 1$.

Next, let's consider the remaining three metrics.

3) Heading similarity. This metric is based on the hypothesis that headings (R^H and C^H) of a source canonical spreadsheet are some classes (types) which generalize entities in a data block (D). First, a set of candidate entities is formed for each cell value v_j from a data block (D). A set of classes C is formed for each entity $e_i \in E$. Next, a set of all values L for heading blocks (R^H and C^H) is formed on the same line as v_j. The similarity between a header value $l_k \in L$ and all classes from C is calculated. In this case, we also use the Levenshtein distance.

Thus, an entity e_i is selected, where $c_j \in C$ for this entity will have maximum similarity for l_k:

$$\psi_3(e_i) = \arg\min HeadingSim\left(l_k, c_j\right), \quad (5)$$

where $\psi_3(e_i)$ is a function for calculating header similarity; l_k is a string k-value of heading from a set of heading values L; c_j is a name of j-class from a set of classes C.

4) Semantic similarity. This metric is based on the hypothesis that data in a source spreadsheet column usually has the same type, i.e. entities usually belong to one class. This is typical for spreadsheets with a common layout in the form of a set of columns with top headers. However, in this paper we consider canonical spreadsheets in which all data from arbitrary spreadsheets is placed in a single column (a block data). Thus, a candidate entity e_i for some cell value v_j must be semantically similar to other entities selected for cell values in a block data (D), and which are contained along with the same heading values in R^H and C^H cells. Therefore, for each entity e_i from a set of candidates E, it is necessary to form a set of classes C to which this candidate entity corresponds. This procedure must be performed for all cell values of a data block (D). Then, the task is to determine the similarity of each class $c_k \in C_l, l = \overline{1,n}$ with classes from other sets of classes $C^{all} = \{C_1, ..., C_n\}$. In this case, we also use the Levenshtein distance:

$$\psi_4(e_i) = SemSim\left(c_k, C^{all}\right), \tag{6}$$

where $\psi_4(e_i)$ is a function for calculating semantic similarity; c_k is a name of k-class from a set of classes $c_k \in C^{all}$; C^{all} is a set of sets of classes for each entity candidate e_i.

5) Mention-entity context similarity. This metric is based on the hypothesis that usually a cell value and a reference entity from a set of candidates have a shared context. First, we select the neighboring cell values by row and column for a certain cell to obtain its context (mention context). Thus, the context for a selected cell value will be a set of all collected cell values. However, this is true only for arbitrary spreadsheets. The definition of context for a cell value v_j from a data block (D) in a source canonical spreadsheet occurs by collecting all cell values for this block corresponding to values from heading blocks (R^H and C^H). These heading values must match heading values located on the same row as v_j. Thus, context for a cell value v_j will be a set of all collected cell values CN^v. RDF triples are collected to define context of an entity e_i from a set of candidates E. In this case, RDF triplets contain this candidate entity e_i. Then, each subject (where a candidate entity e_i is an object) and object (where a candidate entity e_i is a subject,) from these RDF triples is selected. Thus, context for a candidate entity e_i will be a set of all collected entities CN^e. Next, each item from CN^v is mapped to items from CN^e. We use the Levenshtein distance for this mapping. If there is an exact match, then a rank is increased by one:

$$\psi_5(e_i) = \sum_{i=1}^{k} ContextSim\left(cn_i^v, cn_j^e\right), cn_i^v \in CN^v, cn_j^e \in CN^e, \tag{7}$$

where $\psi_5(e_i)$ is a function for calculating mention-entity context similarity; cn_i^v is a string value of i-mention from CN^v; cn_j^e is a name of j-entity from CN^e, $j = \overline{1,n}$.

The aggregated rank can be defined by using all five similarity metrics. This rank represents the final probability that a certain entity e_i from a set of candidates E is referenced (most suitable) for a particular cell value from a data block (D):

$$f_{agg}^D = w_1 \times \left(1 - \frac{\psi_1(e_i)}{100}\right) + w_2 \times \psi_2(e_i) + \tag{8}$$
$$w_3 \times \left(1 - \frac{\psi_1(e_3)}{100}\right) + w_4 \times \psi_4(e_i) + w_5 \times \psi_5(e_i),$$

where f_{agg}^D is a function for calculating the aggregated rank for a candidate entity e_i; w_1, w_2, w_3, w_4, w_5 are weighting factors that balance the importance of ranks obtained using five similarity metrics for disambiguation. All weights are equal to one by default.

Step 4: Generating annotated spreadsheets. At this step, the procedure for generating an annotated spreadsheet with links (tags) to entities from the target DBpedia graph is carried out. An annotated spreadsheet is presented in the Excel format and is available for further processing.

3.3 Implementation

We implemented our approach in the form of two interacting software: TabbyXL [4] and TabbyLD [5]. TabbyXL is used to analyze arbitrary spreadsheets and obtain spreadsheets in the relational canonicalized form. All source arbitrary spreadsheets and canonical spreadsheets are represented in the MS Excel format.

We developed a prototype of software called TabbyLD for the semantic interpretation of data extracted from canonical spreadsheets. TabbyLD has a client-server architecture and is developed using the MVC (Model-View-Controller) pattern, PHP7 and Yii2 framework. This tool consists of four main modules that implement algorithms. TabbyLD has an intuitive graphical user interface and designed for a wide range of users including non-programmers. Canonical spreadsheets in the form of an archive can be loaded, while it is possible to download a single spreadsheet in the MS Excel format. Also, this tool supports a console mode. Results are a set of annotated spreadsheets in the MS Excel format (set on a separate tab in the MS Excel book).

4 EXPERIMENTAL EVALUATION AND DISCUSSION

The purpose of our experiments is to show an applicability of our approach and software for the semantic interpretation of spreadsheet data.

The T2Dv2 Gold Standard [28] is selected as the main dataset for experimental evaluation. The T2D Gold Standard provides a large set of human-generated correspondences between a public Web table corpus and the DBpedia knowledge graph. This dataset consists of manually annotated row-to-instance, attribute-to-property and table-to-class correspondences between 779 Web tables and the DBpedia Knowledge base Version 2014. The tables originate from the English-language subset of the Web Data Commons Web Tables Corpus and cover different topics including places, works, and people. A principle difference to the first version of the gold standard is that T2Dv2 does not only contain positive but also negative examples. Out of the 779 tables in the T2Dv2, 237 tables share at least one instance with DBpedia. We selected all 237 spreadsheets that have correspondences with DBpedia instances, and also 150 negative examples of non-relational spreadsheets for our experiment.

4.1 Evaluation for obtaining canonical spreadsheets

Web tables from T2Dv2 dataset are presented in the JSON format. We developed a JSON2XLSX converter to transform these web tables to the Excel format. Next, according to the approach [23], all obtained spreadsheets containing both positive and negative examples were colorized and converted to a canonicalized form using TabbyXL. Almost ll 387 spreadsheets have been successfully converted to a canonicalized form.

At the stage of obtaining canonical spreadsheets, we used well-known measures to obtain an experimental evaluation such as: precision, recall and F1 score:

$$Precision = \frac{|R \cap S|}{|R|}, Recall = \frac{|R \cap S|}{|S|}, \quad (9)$$

where R is a set of entities in the resulting table, and S is a set of cell values in the corresponding source spreadsheet table.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}, \quad (10)$$

The evaluation results for the stage of obtaining canonical spreadsheets are presented in Table 2.

Table 2: Experimental evaluation for obtaining canonical spreadsheets from T2Dv2

Precision	Recall	F1 Score
0,99	0,97	0,98

High evaluations are obtained through the use of a rule-based approach [23] implemented in the form of a domain-specific language (CRL). An experimental comparison of TabbyXL with two other solutions for understanding tables such as MIPS (Minimum Indexing Point Search) and SENBAZURU is given in [23]. Both solutions aim to transform spreadsheets with a complex structure into a relational (canonicalized) form. However, TabbyXL showed a higher precision, recall and F1 values when processing Troy200 and SAUS datasets.

4.2 Evaluation for annotating canonical spreadsheets

The resulting canonical spreadsheets were annotated using TabbyLD. We used measures of precision, recall, and F1 score for obtaining experimental evaluation at this stage.

$$Precision = \frac{|CAC \cap AC|}{|AC|}, Recall = \frac{|CAC \cap AC|}{|SC|}, \quad (11)$$

where CAC is a set of correctly annotated cell values; AC is a set of annotated cell values (including both positive and negative annotation examples); SC is a total number of cell values in a source canonical spreadsheet.

However, we also used a measure of accuracy in addition to them:

$$Accuracy = \frac{|AC|}{|SC|}, \quad (12)$$

F1 Score is used as a main evaluation. We excluded cells with empty annotations in the file. Weighting factors (w_1, w_2, w_3, w_4, w_5) balance the importance of ranks obtained using five our similarity metrics for disambiguation. So, weighting factors for the first three metrics of disambiguation (string similarity, NER label similarity and heading similarity) were taken by 1. Weighting factors for the remaining two metrics (semantic similarity and mention-entity context similarity) were taken by 1,5, because we proceeded from the hypothesis that these metrics are more important, since they determine what type of cell value belongs to and what context it belongs to. This makes it possible to exclude situations in which the same term may have different semantic meanings. For example, the term "Lincoln" can be the name of a person or the name of a city. The evaluation results for 237 positive examples of canonical spreadsheets are presented in Table 3.

Table 3: Experimental evaluation for annotating positive examples of canonical spreadsheets from T2Dv2

Accuracy	Precision	Recall	F1 Score
0,74	0,73	0,5	0,58

In addition to these 237 spreadsheets, we also selected random 150 negative examples of non-relational spreadsheets that do not have annotations with DBpedia. The accuracy measure for these spreadsheets turned out to be the same as for positive examples. The main purpose of the experiment with negative spreadsheet examples was to demonstrate that our approach and TabbyLD tool are suitable for annotating various spreadsheets, including non-relational ones, for which it is difficult to find annotations in a target knowledge graph.

The total runtime spent for semantic spreadsheet data interpretation amounted to 70275,2748 sec. (19,2 h.).

4.3 Evaluation for Troy200

The Troy200 [29] is selected as the second dataset for obtaining an experimental evaluation. Troy200 contains 200 arbitrary spreadsheets as CSV (Comma Separated Values) files collected from 10 different sources of the same genre, government statistical websites predominantly presented in English language. Troy200 has never been used to evaluate approaches for semantic interpretation of tabular data. Spreadsheets in this dataset contain mainly statistical, numerical information, and are not presented in a relational form. It is also difficult to select annotations with DBpedia for these spreadsheets. However, we decided to apply our approach and software for this dataset, and manually selected and annotated 21 spreadsheets.

The experimental evaluation results for obtaining canonical spreadsheets based on total Troy200 dataset are presented in detail in [23]. We used a measure of accuracy for obtaining experimental evaluation at this stage. Accuracy amounted to 0,64. The total runtime spent on the process of semantic spreadsheet data interpretation amounted to 5010,79 (1,23 h.).

In general, evaluation results showed that our approach can be used for processing various datasets.

5 DISCUSSION

It is worth noting that tables in spreadsheets, such as Excel, come with different formats and layouts. For example, some tables may

include column and row headings while others do not any at all. Our approach is aimed at processing such spreadsheets. However, at the moment, our four similarity metrics, other than the string similarity metric (Levenshtein distance), are focused on processing spreadsheets with headers (column and/or row headers) and do poorly with spreadsheets where there are no headers. These metrics will not work for such spreadsheets. In the future, we plan to improve our approach and develop additional disambiguation techniques to determine the semantic of cell data without column or row headings, or both.

A thorough review indicates that several studies have been performed on the semantic table interpretation. They are primarily aimed at interpretation of tabular data extracted from web pages. However, it is difficult to make a comprehensive quantitative comparison of these researches and our approach, because an experimental evaluation was obtained for various datasets. Despite this, some approaches in recent years have conducted experimented with the T2Dv2 dataset, for example [6, 9, 21]. We considered these approaches as a baseline. Results of a quantitative comparison of precision, recall, and F1 values for annotating positive examples of spreadsheets from T2Dv2 dataset are presented in Table 4.

Table 4: A quantitative comparison of experimental evaluation for annotating positive examples of spreadsheets from T2Dv2 dataset

Approach	Precision	Recall	F1 Score
T2KMatch	0,94	0,73	0,82
TableMiner+	0,96	0,68	0,80
Kruit et al, 2019	0,92	0,86	0,89
Our	0,73	0,50	0,58

Our approach showed an acceptable result at the stage of converting arbitrary spreadsheets to a canonicalized form. However, the result obtained at the stage of annotating canonical spreadsheets turned out to be modest. This is mainly due to the fact that we were not focus on a specific table dataset, unlike other approaches. For example, spreadsheets from Troy200 dataset could not be processed using existing approaches, since these spreadsheets were not presented in relational form. So, we have tried to take the first step towards a universal approach to the semantic interpretation of spreadsheet data.

Main issues resulted to the current score are the following:

- Drawbacks of the algorithm for forming an initial set of candidate entities, which makes recall low. In the future, we plan to use a more flexible algorithm for generating candidate entities, for example, based on Elasticsearch or Solar. We also plan to remove the restriction of 100 candidate entities.
- Drawbacks of the algorithm for annotating data cells with large text (cell values in data and heading blocks are interpreted as a single value). In the future, we plan to divide cell value into parts and annotate these parts separately.
- Inaccurate calibration of metric ranks using weighting factors. In the future we plan to use special models, for example, graph models, Markov chains for this purpose.
- Minor errors during the analysis and transformation of source arbitrary spreadsheets led to the absence of some values in cells for canonical spreadsheets. In our evaluation we identified six such spreadsheets.

6 ACKNOWLEDGMENTS

The contribution of this work was supported by the Russian Science Foundation under Grant No. 18-71-10001.

7 CONCLUSIONS

Semantic interpretation of tabular data stays an area of active scientific research. Existing methods, approaches and tools in this area has limitations and drawbacks both for the considered tabular layouts and for the domains covered.

In this paper, we propose a two-stage approach for semantic interpretation of spreadsheet data presented in the MS Excel format as a first step towards building a universal approach for analyzing and interpreting arbitrary spreadsheets. The main idea of our proposals is to transform source arbitrary spreadsheets to a relational canonicalized form, and use a named entity recognition procedure for each cell of a transformed table. Next, we use five heuristic-based similarity metrics based on the recognized types of named entities for disambiguating and identifying the most suitable (reference) DBpedia entities to linking them with cell values. Our approach is implemented in the form of software: TabbyXL as a tool for analysis and transforming arbitrary spreadsheets and TabbyLD as a tool for annotating canonical spreadsheets.

We used T2Dv2 Gold Standard and Troy200 datasets for experimental evaluation of our approach and software. Experiments have shown their applicability for processing data extracted from arbitrary spreadsheets.

REFERENCES

[1] Michael J. Cafarella, Alon Halevy, Zhe D. Wang, Eugene Wu, and Yang Zhang. 2008. Webtables: exploring the power of tables on the web. In *Proceedings of the VLDB Endowment (VLDB'08).* Auckland, New Zealand, 538–549.

[2] Shuo Zhang, and Krisztian Balog. 2020. Web table extraction, retrieval and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology* 11, 2 (2020), 1-31.

[3] Shuo Yang, Ran Wei, and Alexey O. Shigarov. 2018. Semantic interoperability for electronic business through a novel cross-context semantic document exchange approach. In *Proceedings of the ACM Symposium on Document Engineering (DocEng'18).* Halifax, Canada, 1-10.

[4] Alexey O. Shigarov, Vasiliy V. Khristyuk, and Andrey A. Mikhailov. 2019. TabbyXL: Software platform for rule-based spreadsheet data extraction and transformation. *SoftwareX* 10 (2019), 100270.

[5] *TabbyLD* https://github.com/tabbydoc/tabbyld

[6] Benno Kruit, Peter Boncz, Jacopo Urbani. 2019. Extracting novel facts from tables for knowledge graph completion. In *Proceedings of the 18th International Semantic Web Conference (ISWC'19).* Auckland, New Zealand, 364-381.

[7] Vasilis Efthymiou, Oktie Hassanzadeh, Mariano Rodriguez-Muro, and Vassilis Christophides. 2017. Matching web tables with knowledge base entities: From entity lookups to entity embeddings. In *Proceedings of the 16th International Semantic Web Conference (ISWC'2017).* Vienna, Austria, 260-277.

[8] Basil Ell, Sherzod Hakimov, Philipp Braukmann, Lorenzo Cazzoli, Fabian Kaupmann, Amerigo Mancino, Junaid Altaf Memon, Kai Rother, Abhishek Saini, and Philipp Cimiano. 2017. Towards a large corpus of richly annotated web tables for knowledge base population. In *Proceedings of the 5th International workshop on Linked Data for Information Extraction (LD4IE).* Vienna, Austria, 1-12.

[9] Ziqi Zhang. 2017. Effective and efficient semantic table interpretation using TableMiner+. *Semantic Web* 8, 6 (2017), 921-957.

[10] Tianxing Wu, Shengjia Yan, Zhixin Piao, Liang Xu, Ruiming Wang, and Guilin Qi. 2016. Entity linking in web tables with multiple linked knowledge bases. In *Proceedings of the 6th Joint International Semantic Technology Conference (JIST).* Singapore, Singapore, 239-253.

[11] Mark van Assem, Hajo Rijgersberg, Mari Wigham, and Jan Top. 2010. Converting and annotating quantitative data tables. In *Proceedings of the 9th International Semantic Web Conference (ISWC'10).* Shanghai, China, 16-36.

[12] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. 2010. Converting and annotating quantitative data tables. In *Proceedings of the 36th International Conference on Very Large Data Bases.* Singapore, Singapore, 1338-1347.

[13] Petros Venetis, Alon Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. 2011. Recovering semantics of tables on the web. In *Proceedings of the 37th International Conference on Very Large Data Bases.* Seattle, USA, 528-538.

[14] Varish Mulwad, Tim Finin, and Anupam Joshi. 2012. A domain independent framework for extracting linked semantic data from tables. *Search Computing* 7538 (2012), 16-33.

[15] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q. Zhu. 2012. Understanding tables on the web. In *Proceedings of the 31th International Conference on Conceptual Modeling (ER'12).* Florence, Italy, 141-155.

[16] Wei Shen, Jianyong Wang, Ping Luo, and Min Wang. 2012. LIEGE: Link entities in web lists with knowledge base. In *Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'12).* Beijing, China, 1424-1432.

[17] Dong Deng, Yu Jiang, Guoliang Li, Jian Li, and Cong Yu. 2013. Scalable column concept determination for web tables using large knowledge bases. In *Proceedings of the 39th International Conference on Very Large Data Bases.* Riva del Garda, Trento, Italy, 1606-1617.

[18] Emir Muñoz, Aidan Hogan, and Alessandra Mileo. 2014. Using linked data to mine RDF from wikipedia's tables. In *Proceedings of the 7th ACM international conference on Web search and data mining.* New York, USA, 533-542.

[19] Chandra S. Bhagavatula, Thanapon Noraset, and Doug Downey. 2015. TabEL: Entity linking in web tables. In *Proceedings of the 14th International Semantic Web Conference (ISWC'15).* Bethlehem, USA, 425-441.

[20] Ivan Ermilov, and Axel-Cyrille N. Ngomo. 2016. TAIPAN: Automatic property mapping for tabular data. In *Proceedings of the 20th International Conference on European Knowledge Acquisition Workshop (EKAW'16).* Bologna, Italy, 163-179.

[21] IDominique Ritze, and Christian Bizer. 2017. Matching web tables to DBpedia - A feature utility study. In *Proceedings of the 20th International Conference on Extending Database Technology (EDBT'17).* Venice, Italy, 210-221.

[22] *SemTab: Semantic Web Challenge on Tabular Data to Knowledge Graph Matching* https://www.cs.ox.ac.uk/isg/challenges/sem-tab/

[23] Alexey O. Shigarov, and Andrey A. Mikhailov. 2017. Rule-based spreadsheet data transformation from arbitrary to relational tables. *Information Systems* 71 (2017), 123-136.

[24] Michael J. Cafarella, Alon Halevy, Daisy Z. Wang, Eugene Wu, and Yang Zhang. 2008. Uncovering the relational web. In *Proceedings of the Eleventh International Workshop on the Web and Databases (WebDB'08).* Vancouver, Canada, 1-6.

[25] *Stanford CoreNLP* https://stanfordnlp.github.io/CoreNLP/

[26] *Stanford CoreNLP: Named Entity Recognition* https://stanfordnlp.github.io/CoreNLP/ner.html

[27] AChristian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - A crystallization point for the Web of Data. *Journal of Web Semantics* 7, 3 (2009), 154–165.

[28] *T2Dv2 Gold Standard for Matching Web Tables to DBpedia* http://webdatacommons.org/webtables/goldstandardV2.html

[29] *TANGO-DocLab web tables from international statistical sites (Troy*$_2$00) $http://tc11.cvc.uab.es/datasets/Troy\_200\_1$

Speculative Query Execution in RDBMS Based on Analysis of Query Stream Multigraphs.

Anna Sasak-Okoń
University of Maria Curie-Skłodowska
5 Maria Curie-Skłodowska Sq.
Lublin, Poland
anna.sasak@umcs.pl

Marek Tudruj
Institute of Computer Science
Polish Academy of Sciences
5 Jana Kazimierza Str.
Polish-Japanese Academy of Information Technology,
83 Koszykowa Str.
Warsaw, Poland
tudruj@ipipan.waw.pl

ABSTRACT

The paper presents an insight into a speculative execution model of queries in RDBMS based on the analysis of the stream of current queries appearing at the database input. A specific multigraph representation of input query stream is created and used to determine the speculative queries for execution. A group of worker threads execute the chosen speculative queries in parallel with the execution of the standard input stream of user queries. The obtained speculative results are then used to support faster query execution. First, the paper briefly reminds the assumed graph modelling and analysis methods. Then, additional rules are presented which enable combining results of multiple speculative queries in execution of a single user input query. The quality of executed and used speculations is then analysed based on the defined quality metrics and structural details of speculative queries. Conclusions from this analysis are used to modify the selection method of target queries for speculative execution. It aims at intensification of the use of multiple speculative query results and further reduction of the user query execution time. Experimental results are presented in a multi-threaded speculative experimental environment cooperating with a SQLite database. They show that with the improved algorithm we can obtain more varied speculative query results, and thus, more intensive use of multiple speculative query results by the stream of user queries sent to the database.

CCS CONCEPTS

• **Information systems → Query optimization**; • **Computing methodologies → Parallel computing methodologies**.

KEYWORDS

speculative computations, speculative database queries, query graph modelling, relational databases

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06.
https://doi.org/10.1145/3410566.3410604

ACM Reference Format:
Anna Sasak-Okoń and Marek Tudruj. 2020. Speculative Query Execution in RDBMS Based on Analysis of Query Stream Multigraphs.. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3410566.3410604

1 INTRODUCTION

The beginnings of the speculative program execution goes down to branch prediction technology included in processor architecture [3][4]. The idea was to execute all conditional branches in parallel with computations which produced the control data for the choice. Predicting which branch of control statement should be chosen before having all the control data is called branch speculation [5]. The approach described as a speculative decomposition [2] allows for parallelization of essentially serial programs at a cost of the use of some extra computational power. Comprehensive surveys on speculative execution can be found in [1][6].

In the literature, the practical use of the speculative approach in relational data bases concerns mainly supporting single queries or transactions by performing operations such as execution of chosen subqueries in advance, out of the standard order. In this work, we concentrate on parallelized speculative support for execution of streams of queries. The speculative support model proposed in our previous papers [30][31] assumes execution of additional Speculative Queries which are chosen based on an automatic analysis of the stream of queries arriving to a Relational Database Management System. In our approach, we have introduced a query window, called the Speculation Window, moving on the input stream of user queries. It is a base for the analysis which determines the consecutive steps of the speculative support in the query stream execution. This analysis is done by a multithreaded middleware called the Speculation Layer, situated between user applications and the RDBMS. The Speculation Layer combines the relevant single query graph representations into one multigraph according to the defined set of rules. The combined multigraph contains an union of vertices and a multiset of edges of all relevant query graphs which contains equivalents of all edges which appeared in the combined query graphs.

The proposed Speculative Query execution support is directed into data applications which usually execute queries of a similar type (e.g. getting information on a product according to specified criteria) and where data modifications are relatively rare. For now,

the Speculative Layer is resilient to unexpected database contents changes. A more sophisticated reaction mechanism is not yet included.
The stream of user queries waiting for execution has a few typical features:

- **numerous** – the number of incoming queries is large enough to form a queue,
- **parametric**– each query follows an expression pattern from the group of allowed patterns with varying values depending on users activity,
- **growing**– the stream expands by queries added by users with different frequencies.

Based on the aforementioned multigraph representation, the Speculative Queries are determined as a result of the Speculation Window Query Multigraph analyses. Starting from the attribute vertices, all possible condition values are considered, which are then used to create Speculative Queries. The results of executed Speculative Queries are then stored in the main memory structures called Speculative DB. This set of data provide quick access to a specific selection of rows and/or columns which could be useful for possibly biggest number of queries in the customer query stream.
The Speculative Layer algorithms are based on two kinds of Speculative Queries performed by threads of the RDBMS server:

- The first kind includes in advance generated Speculative Queries. A joint graph representation of queries covered by the Speculation Window is used to detect common subexpressions and to generate possibly the most useful Speculative Queries.
- The second kind is generated on demand when some of RDBMS server threads become idle. The on demand speculative queries consider the execution history of previous customer queries, which is continually registered by the Speculative Layer.

In our previous work [30][31], a simplified speculation model was used which assumed the use of the results of one best Speculative Query for each input query. This paper presents more detailed analysis of the extended Speculative Layer functionality presented in [32]. We have expanded the Speculative Layer functionality by allowing the use of the results of multiple Speculative Queries to support a single input query. Now, the relations in an input query can be speculatively supported in parallel by different speculative queries giving a significant reduction of query response time. The proposed speculative support has been positively verified by extensive tests performed in the SQLite environment. More than 80% of test queries were executed with the use of at least one Speculative Query result. Paper [32] presents only basic analysis of how many queries were executed with the use of multiple Speculative Queries. In this paper, we also present a quality and performance analysis of the executed Speculative Queries, based on defined metrics and the type of the operator used in query structures. This analysis is used to propose a new version of the algorithm for choosing Speculative Queries to be executed. As a consequence, a new group of Speculative Queries appeared, and more user queries were executed using more than one Speculative Query result.
The remaining text of the paper is composed of 8 parts. In the first part the related work is described. Next, a general structure of the proposed database framework is presented. Two parts that follow describe the assumed query structures and the graph representation used to define Speculative Queries. Next, two metrics are defined that are used to choose and then validate the use of speculative queries. Section 7 introduces the algorithms used in the proposed speculative database support. First, for multiple speculative queries use and then for restricted speculation choice. The rationale for the decision on the size of the analysed query window size is then presented. The results of the experiments which assess the proposed solutions are presented in the final two parts, extended with new results concerning the defined quality metrics and query structure analysis.

2 RELATED WORK

In general, a speculative action is considered as some work done in anticipation of an event to take place. This may end with a gain or loss in application execution time that depends on speculation accuracy [11]. The model of speculative execution has been developing predominantly while looking for new levels of parallelism. Speculative execution has found especially favourable ambience with the advent and burst development of the thread level parallelism [12]. The thread level parallelism is fundamental in the context of the speculative parallelization. Three main concepts of thread level speculations have been distinguished in the literature: control speculation, data value speculation and data dependence speculation [1][13][14][15]. Control speculation implies the execution of an instruction before the execution of a preceding instruction on which it is control dependent. In data value speculation an attempt is made to predict the data value that an instruction is going to produce. In data dependence speculation, no explicit attempt is made to predict data values. Instead, a prediction is made on whether the input data value of an instruction has been generated and stored in the corresponding named location (memory or register) [16].
We will now survey some most relevant works on employing speculative execution in databases. Proposals of applying speculative execution aiming in integration of data coming from separate heterogeneous database sources (data gathering plans) were published in [9][10]. Based on data hints received earlier in the data gathering plans, some operations were performed speculatively, ahead of their normal schedule. Due to some operations executed in parallel (speculatively), significant plan implementation speedups were obtained, when predictions were correct. A proposal of using speculation to support SQL query processing was presented in [7][8]. It was based on an idea of using the database system idle time for asynchronous anticipated database data transformations performed in a parallel way. In case, when the lookahead performed operations were useful for the final query, the target query was executed in a shorter time.
Using speculative execution to support transaction protocols in databases was proposed in [17][18][19]. An idea of the speculative protocol has been described which enables a faster access to data locked in a transaction. Two speculative executions based on old and recent transaction images were performed. Finally, one of the speculative results was validated depending on the obtained blocking transaction real result.

A proposal of using speculative computations in records analysis for ranked queries has been presented in [11]. The queries aimed in returning records according to some preference function, when some inaccuracy of results was enabled. The method assumed creating speculative versions of the ranking algorithms. It was possible to speculatively assume the degree of query results variation as negative in the case of a slower data source. Due to this, the query results could be returned faster but with a risk some approximative character of the results.

The techniques to support multiple queries optimization, though without use of the speculation concept, are proposed for example in [20][21]. If a group of queries share common sub-expressions they can be used either to choose more appropriate execution plan or as candidates for potential materialized views. In both cases the total execution time is decreased by performing common tasks only once. Very often there is a need for a formalism that can be used to represent analysed queries, which in many cases are graphs [22][23][24]. Graphs are ubiquitous as they naturally model entities and their relationships.

A popular method of query optimization is cashing the results of previously executed queries. It assumes that there is a relatively high probability that the same query will be requested again, thus instead of evaluating the query the cached result can be reused [26][27][28] [29]. With the use of three cash variants: tuple cache, page cache, and semantic cache, the target query is decomposed in such a way that only data not available in cashes are loaded from the server. The proposed algorithms attempt to find the reusable components of earlier query plans, and then,they develop a new query plan to reuse these components. While the resulting query plans may not be optimal, they try to enable more data reuse, and hence, to speedup execution.

It should be emphasised that none of the discussed above query optimization methods was targeting a co-operative definition of the speculative actions that would cover needs of many future queries at a time. Instead of generating a query plan for a new query based on a set of similar previous queries, we analyse queries waiting for execution in the nearest future and support these queries by using data prepared for them in advance as a result of new specially defined Speculative Queries. This feature together with a multigraph-oriented approach are essential originality features of our method.

3 A SQL DATABASE WITH A SPECULATIVE LAYER

Analysing the stream of consecutive queries an observation can be easily made that they are similar to a set of successive instructions in a sequential computer program. What's more, the consecutive queries can also contain some common constituent operations. Based on these observations, a parallel approach similar to sequential program parallelization, can be applied to SQL queries. If only a RDBMS provides a relatively high level of data stability, partial results of equivalent operators in some queries can be used many times by currently awaiting other queries.

In our previous papers [30][31], we have described a model of Speculative Query execution which is implemented as an additional

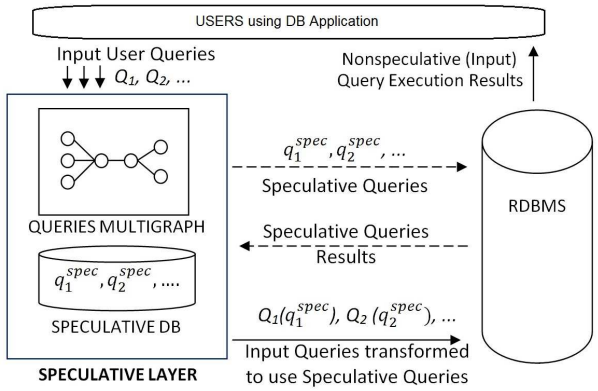


Figure 1: RDBMS with an additional Speculative Layer analysing user queries.

middleware called the Speculative Layer, located between user database applications and the RDBMS. It provides a support for execution of user queries and thus shortens the query response time. The general scheme of the RDBMS cooperating with the Speculative Layer is presented in Fig.1.

The main task of the Speculative Layer is the analysis of the current stream of the user queries. First, it creates separate representations of queries from the Speculation Window. Then, these representations are combined into a single multigraph which is analysed to determine a set of Speculative Queries to be executed. A set of data containing the results of the Speculative Queries is called a Speculative Database (Speculative DB) and it is stored in the RDBMS server main memory. The data from the Speculative DB can (but doesn't have to) be used during execution of customer queries. Speculative Queries provide ready to use working data in the main memory, and if used, they eliminate scanning of large numbers of records which improves system throughput and shortens user waiting time. On the other hand, queries executed in a speculatively non-supported way are usually forced to perform a full table scan with the use of slow disk memory transactions.

The Speculative Layer is initiated and then controlled by the main worker thread called the Manager. It coordinates and supervises the currentness of the graph representations. At the beginning, graph representations are created for a first group of queries from the Speculation Window. Next, these representations are merged together in a specific way to create the Query Multigraph ready for the Speculative Analysis. As a result of this analysis a Speculative Query Multigraph is generated. It contains new type of edges - the speculative edges - which determine the type and the contents of the Speculative Queries that can be executed. Two validation metrics are defined which are then used to choose the Speculative Queries to be executed with the results stored in the Speculative DB(such queries are called Awaiting Speculative Queries). The executed Speculative Queries are then assigned to the selected user query or queries from the Speculation Window, to enable using these Speculative Query results. After the first user input query in the Speculation Window is executed (possibly with the use of Speculative Query results), its results are returned to the user and

the Speculation Window moves on by one user query in the input queue. As a consequence, the representation of the executed input query in the Query Multigraph is replaced by the representation of the next user query from the queue. The process repeats as long as there are user queries waiting to be executed. When the Speculative DB reaches its maximum size, it has to be reduced. The reduction process consists in removing the results of chosen executed Speculative Queries based on their characteristics. First to remove are queries with the highest Vertical and Horizontal Selectivity and those that were used the least often. For a detailed description of the Speculative Layer functions including the Speculative Graph Analysis and Queries Execution see [30][31].

4 ASSUMED QUERY STRUCTURE AND GRAPH REPRESENTATION

The Speculative Layer supports the CQAC (Conjunctive Queries With Arithmetic Comparisons) queries. They are SPJ (Selection - Projection - Join) queries with one type of logical operator - AND - in WHERE clauses. Additionally there are two more operators allowed: IN for value sets and LIKE for string comparisons and a nested query which can appear in the WHERE clause. For the purpose of this article we also assume that each input query relates to at least 2 different relations. Each CQAC query is represented by its Query Graph $G_Q(V_Q, E_Q)$ according to rules similar to these proposed in [35][36].

In this paper we assume two types of user queries: the standard queries (SELECT) and the modifying queries (MODIFY) which introduce modifications into the considered database. There are three types of vertices in the graphs of the two assumed types of queries:

- a Query Relation - one for each relation from the WHERE clause of the represented query,
- a Relation Attribute - one for each attribute from the query,
- an Attribute Value - one for each value from the represented query WHERE clause.

Query Graph edges represent relations between adjacent vertices resulting directly from the structure of the represented query. For the standard (SELECT) type queries we have the following types of edges:

- a Membership Edge - μ - between a database relation vertex and each of its attributes from the SELECT clause in the represented query,
- a Predicate Edge - θ - between two attribute vertices or an attribute vertex and an attribute value vertex for each predicate of the query WHERE clause,
- a Selection Edge - σ - between a query relation vertex and an attribute vertex, one for each predicate of the query WHERE clause.

For the modifying (DELETE, UPDATE, INSERT) type queries we have the following types of edges placed between a database relation vertex and a modified attribute vertex:

- a Delete Edge δ,
- an Insert Edge η,
- an Update Edge υ.

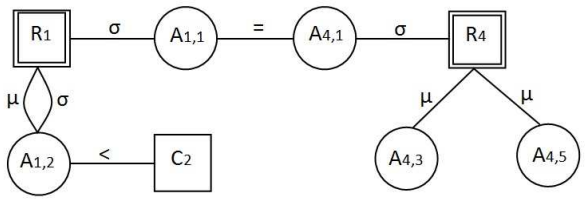


Figure 2: Single query graph representation.

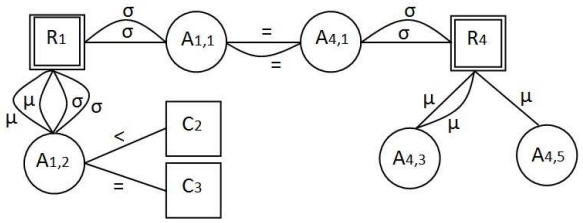


Figure 3: Query Multigraph.

Graph representation of the following query: SELECT $A_{1,2}, A_{4,3}, A_{4,5}$ FROM R_1, R_4 WHERE $A_{1,1} = A_{4,1}$ AND $A_{1,2} < C_2$ is shown in Fig.2.

5 QUERY MULTIGRAPH AND SPECULATIVE ANALYSIS

A set of queries is represented by a single graph $G_S(V_S, E_S)$ called a Query Multigraph or QM. QM vertices set is an union of vertices of all component query graphs: $V_s = V_{Q_1} \cup V_{Q_2} \cup ... \cup V_{Q_n}$. QM edges set is a multiset of all component query graph edges: $E_s = E_{q1} + E_{q2} + \ldots + E_{qn}$. This way multiple edges of the same type are allowed raising the issue of some edge grouping.
Fig.3 shows the QM representing the following component queries: (SELECT $A_{1,2}, A_{4,3}, A_{4,5}$ FROM R_1, R_4 WHERE $A_{1,1} = A_{4,1}$ AND $A_{1,2} < C_2$) and (SELECT $A_{1,2}, A_{4,3}$ FROM R_1, R_4 WHERE $A_{1,1} = A_{4,1}$ AND $A_{1,2} = C_3$).

Speculative Analysis is a process of inserting Speculative Edges in the QM which corresponds to the Speculation Window as an indication for respective Speculative Queries to be generated. In Fig.4 they are denoted by dashed lines. These edges correspond to different strategies to be undertaken for creating the assumed kinds of Speculative Queries.
Based on the desired speculation results, we have distinguished three types of Speculative Queries (edges):

- **Speculative Parameter Queries** - the inserted speculative edges mark selected nested queries for speculative execution. If a nested query has been marked to become a Speculative Query, it is possible to use its results as a parameter in its parent query.
- **Speculative Data Queries** - the aim of these Speculative Queries is to obtain and save in the Speculative DB a specific subset of records or/and attributes of a relation so as it could be used while executing as many user queries as possible.

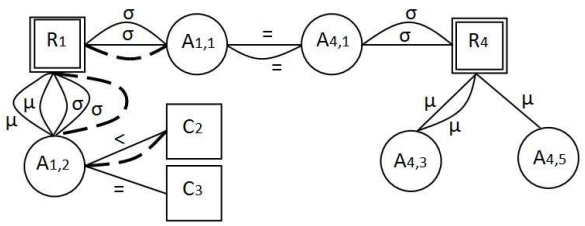


Figure 4: QM with speculative edges representing a single Speculative Query.

- **Speculative State** - the inserted edges refer to modifying queries. If there are modifying queries in the Speculation Window then both already executed and Awaiting Speculative Queries are in danger of processing invalid data so some specific actions have to be performed. After the threatening modifying query is executed all the marked Speculative Queries of the Speculative State type are verified to check if the modifying query has influenced their results in the way that the speculation has no longer valid results.

The algorithm of inserting Speculative Queries is executed for each attribute vertex incident to a selection edge. A nested query can be identified resulting in a new Speculative Parameter Query. If that does not occur, the best Speculative Data Query for this vertex is determined. All combinations of value vertices adjacent to the analysed attribute vertex are considered to form its WHERE clause. For a chosen attribute vertex and the set of value vertices extended with adjacent relation vertex and additional attribute vertices (for SELECT clause), Speculative Data edges are inserted into the QM. If a modifying query appears in the position K of the Speculation Window, then all succeeding queries (denoted by us as K+) are in danger of processing invalid data. Thus, all Speculative Parameter and Speculative Data Edges corresponding to K+ queries must be marked by additional Speculative State edges. The detailed algorithms for inserting speculative edges into the QM and generating Speculative Queries are described in [31]. Fig. 4 shows the QM from the Fig. 3 with speculative edges representing one Speculative Query: SELECT $A_{1,1}, A_{1,2}$ FROM R_1 WHERE $A_{1,2} < C_2$. Assuming that $C_3 < C_2$ this Speculative Query results can be used by both queries represented by the analysed QM.

6 VALIDATION AND USE OF SPECULATIVE QUERIES

As a result of the Speculative Analysis new Speculative Queries are created and appended to the Awaiting Speculative Query List which is kept in the main memory. The awaiting Speculative Queries execution order should not be random but as optimal as possible. It means that the highest execution priority should belong to such queries which provide the highest reduction of the number of records or/and can be used by the biggest number of input queries. To assess the Speculative Queries priorities two main metrics are defined and used: Vertical and Horizontal Selectivity.

Vertical Selectivity of a Speculative Query Q_s for the R_i relation corresponds to the reduction of the number of columns implied by the query and is expressed as follows:

$$\gamma_V(Q_s, R_i) = \frac{size_{row}(R_i) - size(\pi_s)}{size_{row}(R_i)} \tag{1}$$

where $size_{row}(R_i)$ stands for the row size of the relation that the query is referring to and $size(\pi_s)$ is the size of all attributes from SELECT and WHERE clauses of Q_s.

Horizontal Selectivity of a Speculative Query Q_s for the R_i relation corresponds to the number of data records according to the Q_s WHERE clause and is expressed as follows:

$$\gamma_H(Q_s, R_i) = \frac{\omega_\theta(Q_s, Val)}{count(R_i)} \tag{2}$$

where $\omega_\theta(Q_s, Val)$ approximates the number of records consistent with Q_s WHERE clause and count(R_i) is the total number of R_i records.

$$\omega_\theta(Q_s, Val) = \begin{cases} \frac{1}{Val_{max} - Val_{min}}, \theta \equiv = \\ \frac{Val_{max} - Val}{Val_{max} - Val_{min}}, \theta \equiv > \\ \frac{Val - Val_{max}}{Val_{max} - Val_{min}}, \theta \equiv < \\ \omega_>(Q_s, Val) + \omega_=(Q_s, Val), \theta \equiv \geq \\ \omega_<(Q_s, Val) + \omega_=(Q_s, Val), \theta \equiv \leq \end{cases} \tag{3}$$

where Val is the value from the analyzed condition in the WHERE clause, Val_{max} and Val_{min} are the maximal and the minimal values of the attribute from the analysed condition. For two special operators - LIKE and IN, the Horizontal Selectivity is expressed as follows:

$$\omega_\theta(Q_s, Val) = \begin{cases} \frac{1}{ValDist(R_i, null, A_j)}, LIKE \\ \frac{\overline{Val}}{ValDist(R_i, null, A_j)}, IN \end{cases} \tag{4}$$

where
$ValDist(R_i, null, A_j) = Res('SELECT count(distinct A_j) FROM R_i')$
and $\overline{Val}$ is the number of the elements in the set the IN operator is referring to.
Horizontal Selectivity provides good approximation of the number of records returned by Q_s since data used for experiments have a uniform attribute values distribution. Both defined metrics are used to classify the awaiting Speculative Queries and to determine which of them should be executed with the highest priority.

7 DUAL-SPECULATION TYPE ALGORITHM

In our previous papers [30][31] we proposed a dual-speculation type algorithm in which two kinds of Speculative Queries are used: and Speculative Queries on Demand. The Speculative Queries in Advance are generated based on the multigraph representation of queries from the current Speculation Window. The number of such queries is limited to 2 (in the previous version of this algorithm it was unlimited). The Speculative Queries in Advance have the following features:

- The first Speculative Query in Advance corresponds to the highest number of input queries which could use its results,
- The second Speculative Query in Advance has the highest row reduction count for all speculative queries generated for the analysed Speculation Window (estimated by their Horizontal Selectivity).

The limited number of Speculative Queries in Advance provides a chance of having idle threads which execute the Speculative Analysis, so we decided to use a spare computing power (if available) to execute the second type of Speculative Queries - the Speculative Queries on Demand. These queries are generated only when a worker thread sends a no-job request. They are based on the analysis of the history of previously executed queries assuming the following rules:

- a single Speculative Query on Demand is generated for the attribute which the most frequently occurred in the registered history of user query executions but hasn't been used in any Speculative Query on Demand execution yet.
- if there already exist Speculative Queries on Demand for all attributes in the registered history as above, the Speculative Query on Demand is generated referring to the attribute with the highest occurrence rise coefficient since the previous Speculative Query on Demand execution.

In our previous works, we assumed that each executed input query could use results of only one Speculative Query on Demand or in Advance. In paper [32] we made a first attempt to use the results of multiple Speculative Queries by a single input query. The multiple use of Speculative Queries combined with a strategy for the modifying query detection was presented in [33]. In this paper we provide a more detailed analysis of the executed and used speculations and on how they affect the executed user queries. Previously, the basic criteria for a Speculative Query and user query evaluation was their execution time, and how it was affected by the use of Speculative Queries. Now, we additionally present detailed characteristics of executed Speculative Queries concerning not only values of defined metrics (Vertical and Horizontal Selectivity) but also which database relations are referred with what kinds of operators. Based on this analysis, we amend the algorithm of choosing the speculative queries for execution. So far, the priority of execution was always given to the Speculative Queries which had chance to be used by the highest number of user queries from the Speculation Window. Now, we deny execution of Speculative Queries which have the Horizontal Selectivity over 0.8, and additionally we add a preference flag to mark queries with the Horizontal Selectivity smaller than 0.1.

Speculation-supported execution of an input user query can be illustrated by the following pseudo code:

```
ExecuteUserQuery(Q={R1,R2,...,Rn}){
//where Rn are relations occurring in an input query Q
  foreach ( Ri IN Q){
   foundSQ <- specForQ.find(Q.Ri)//in the current Spec.
    //Query List find the best query referring to Ri
    // and register in foundSQ
    if (foundSQ not null)
      Q.query <- adaptquery(foundSQ)
      //edit Q to use the foundSQ result
  }
  execute(Q.query)
  return;}
```

A procedure for choosing a speculative query to be executed by a worker thread can be described by the following pseudo code:

```
ChooseSpeculation(QS={QS1, QS2,...,QSm}){
//where QS lists the Speculative Queries QSn generated
// based on the current Speculation Window
  QS_most_usefull = empty_query;
  //QS_most_usefull indicates the best QSi
  QS_most_usefull.num = 0;
  //user query # using QS_most_usefull query
  QS_smallest = empty query;
  // Spec. Query with the smallest Horiz. Selectiv. HS
  QS_smallest.HS= 0.8;
  foreach ( qi IN QS){
    if qi.Horiz.Select.> 0.8 ignore_query
    else { if (qi.num > QS_most_useful)
             QS_most_usefull = qi;
           if (qi.HS < QS_smallest) QS_smallest = qi;}
   }
  if (qs_smallest.HS<0.1) return qs_smallest
  else return qs_most_usefull;  }
```

The Speculative Layer was implemented in C++ and Visual Studio 2013 with Pthread library and SQLite 3.8.11.1. The experimental results were obtained under Windows 8.1 64b with Intel Core i7-3930K processor and 8GB RAM. For the experiments we used the database structure and data (8 relations, 1GB of data) from the TPC benchmark described in [34]. However, a new set of 9 Query Templates was prepared and used to generate 3 sets of 1000 input queries each:

- Templates T1-T8 joins at least 2 relations and occurs in the test set of 1000 queries with the same density of 12 %.
  - T1-T4 join two different relations each,
  - T5, T6 join three relations each,
  - T7, T8 join four relations,
  - T8 joins five relations.
- T9 template is an example of modifying query with the density of 4 %.

Thus, the presented results for each template T1-T8, such as query execution time, are the average values of execution times of all queries of a particular template from 3 test sets, which were computed for approximately 360 queries of each type. Every value used in the WHERE clause was a proper random value for the attribute it referred to. Some of the WHERE clause attributes are shared by different templates which encourages the use and execution of Speculative Queries. For templates T9 no Speculative Queries were computed. T9 queries were used only to introduce a risk in using the existing Speculative Query results. Structures of the T1-T9 templates are presented below with the following notation:
TemplateName: RELATIONname(attributes of the WHERE clause), ..., RELATIONname(attributes of the WHERE clause).
T1: LINEITEM (discount, qty), PART (brand, container)
T2: PART (brand, type, size), PARTSUPP (availqty)
T3: ORDERS (total, priority), CUSTOMER (segment)
T4: LINEITEM (orderkey), ORDERS (orderdate, orderkey)
T5: LINEITEM (price, qty), ORDERS (total, priority), CUSTOMER (segment)

T6: LINEITEM (discount, qty), PART (brand, type, size), PARTSUPP (availqty)
T7: LINEITEM (price, qty), ORDERS (priority), CUSTOMER (segment), PART(type, size)
T8: LINEITEM (price, qty), ORDERS (priority), CUSTOMER (segment), PART (type, size), PARTSUPP (availqty)
T9: UPDATE ORDERS(total, priority)

8 EXPERIMENTAL RESULTS

8.1 Speculation Window Size

At the beginning a series of experiments was conducted to determine the size of Speculation Window which stands for the number of input queries represented by QM. Based on obtained results which are presented in [32] it was decided that further experiments will be carried out for the Speculation Window size equal to 5.

8.2 Multiple Speculative Queries

After the Speculation Window size was determined, a new experiment was conducted. Based on this experiment the number of active threads (i.e. threads which execute Speculative Queries) was set to 3. The detailed results are presented in [32]. Then, a final set of experiments was conducted to investigate the utilization of multiple Speculative Queries by single user queries. Fig.5 and Fig.6 present results we obtained for the Speculative Window size equal to 5 and with 3 worker threads used. Both charts present the results for different query Templates. We show results for only T1-T8 templates since T9 is a modifying query template which can't be executed with the use of speculation and only affects the usability of already executed Speculative Queries. It can be noticed, that a satisfactory percent of input queries used results of multiple Speculative Queries, while only about 7% of input queries were executed without the use of any Speculative Query result (Fig.6). As a consequence, a distinct reduction of the average execution time for each template can be observed. Each additionally used speculative query provides further average execution time reduction: from 10% (T7 - the second and third Speculative Queries) to 70% (T2 - the second Speculative Query), see Fig.7.

The results obtained and presented in [32] encouraged us to a more detailed analysis which could lead to an improvement which has increased the number of queries which used more than one Speculative Query. Except the T1 template, all other templates (especially T3) could benefit from such algorithm improvement.

8.3 Quality of Speculations

According to the metrics defined in Section 6 we present a brief analysis of the executed Speculative Queries. A quality of a Speculative Query described by its vertical and horizontal selectivity is an important factor influencing the effectiveness of the Speculative Layer.

Fig.7 presents what values of the Vertical Selectivity were obtained for the executed Speculative Queries. In this figure, the Vertical Selectivity represents the reduction of the number of columns from the original relation. We can observe that about 40% of speculations, has the Vertical Selectivity close to 0,7 (parts of the chart marked with values 13,9%, 16,5% and 14,2%), which provides 30%

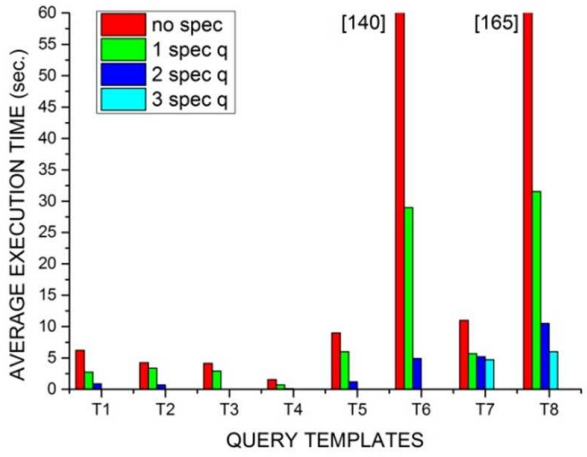


Figure 5: The average execution time with the use of multiple Speculative Query results [32].

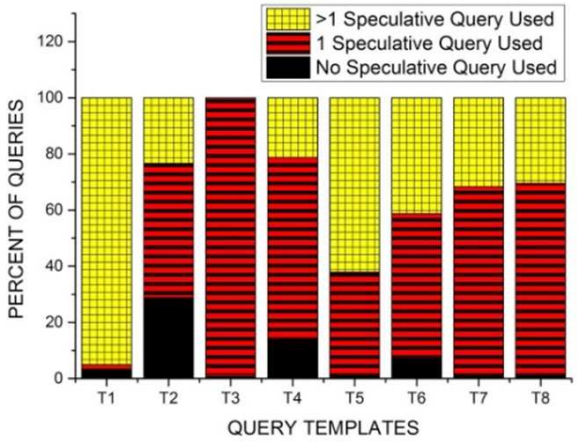


Figure 6: The percent of input queries which used the Speculative Query results [32].

less columns in the Speculative DB than in the original DB. The less columns we have to copy, the less space is needed for a particular Speculative Query results. On the other hand, the values of the Vertical Selectivity show very weak algorithm dependence, as the list of columns is usually fixed (or at least partially fixed) for a particular query type, not allowing for a big reduction.
Fig.8 shows the values of the Horizontal Selectivity of the executed Speculative Queries. These values are very important as they show the reduction in the number of rows copied from the main database relations to the Speculative DB. Almost 70% of the executed Speculative Queries had the Horizontal Selectivity smaller than 0,1 (part of the chart marked as 68,4%). That proves a fair effectiveness of the dual-speculation type algorithm as about two third of executed speculations copied less than 10% of rows from the original relation. However, one should pay attention to the Speculative Queries with the Horizontal Selectivity equal to 1, which means the copying of all records from the main database done, and the only size reduction could come from the Vertical Selectivity. There

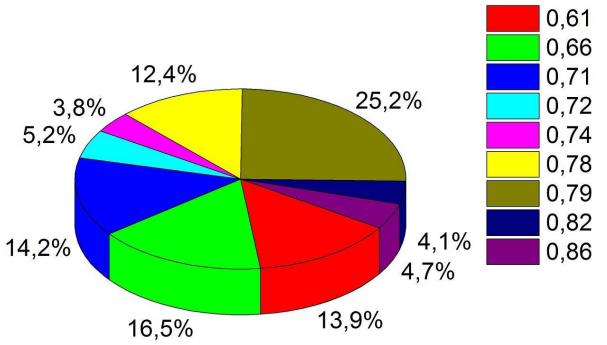


Figure 7: The percent of Speculative Queries with the estimated values of the Vertical Selectivity.

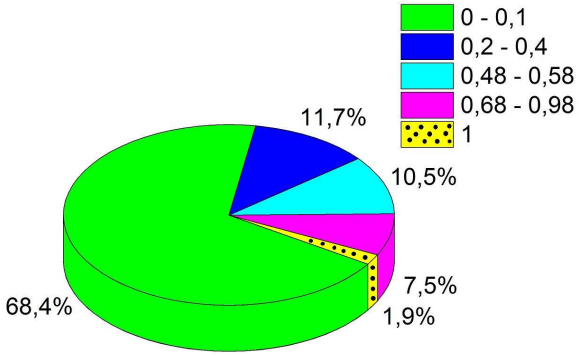


Figure 8: The percent of Speculative Queries with the estimated values of the Horizontal Selectivity.

were only about 2% of such Speculative Queries but even this percentage of queries is not desirable, as the more space those queries are occupying in the Speculative DB, the more often we have to execute the process of Speculative DB cleanup to free space for new speculation results.

Fig.9 presents which relations were referred to by executed Speculative Queries and how often. We can see that only three out of five relations from query templates are represented in executed Speculative Queries. Furthermore all Speculative Queries referring to the Orders relation, showed the Horizontal Selectivity ranging from 0,01 to 0,02, and for the Part Relation in the range 0,01 to 0,1. Special attention should be paid to the speculations referring to the Lineitem relation which is also the biggest in the database. As both Orders and Part speculations show small values of the Horizontal Selectivity, all "heavy" speculations (with the Horizontal Selectivity close to 1) refer to the Lineitem relation making it less profitable. We can assume these would be the speculations with the execution time close to the maximal value, presented in Table 1).

From Fig.9 we know that over 50% of Speculative Queries are referring to the biggest relation in the database which is LINEITEM. Fig.10 presents an insight to the Speculative Query structure, in

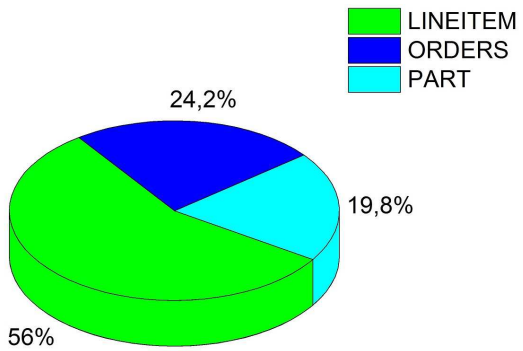


Figure 9: The percent of Speculative Queries executed for each of the three database relations without restrictions.

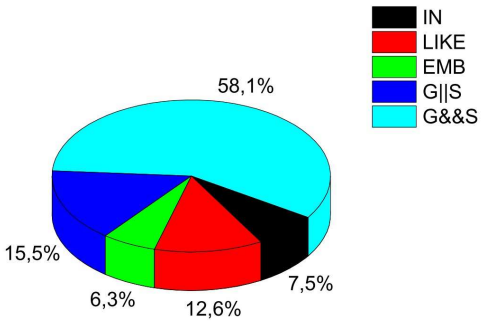


Figure 10: The operator type used in the executed Speculative Queries in the algorithm without restrictions.

Table 1: Average, maximal and minimal execution times of Speculative Queries for each database relation.

	EXECUTION TIME (sec.)		
DB RELATION	**AVERAGE**	**MAXIMAL**	**MINIMAL**
LINEITEM	6,056	12,186	4,337
ORDERS	0,973	1,556	0,825
PART	0,186	0,375	0,132

particular which type of operator is used and how it affects its effectiveness (Table 2). We can see that there are five types of Speculative Queries. First two groups are IN and LIKE queries. If a query is built with one of these operators it has low values of the average Horizontal and Vertical Selectivities and a short average execution time (Table 2). Both groups represent around 20% of executed Speculative Queries. Next group is composed of queries marked with EMB label i.e. queries that are the exact copies of an embedded query found in a user query. Even though these queries structure is algorithm independent, they also present low values of parameters presented in Table 2. The most interesting are two last groups of Speculative Queries representing over 70% of all executed Speculative Queries. Label $G||S$ means that these queries used an unique limitation of the attribute value (i.e < or >) while the label $G\&\&S$

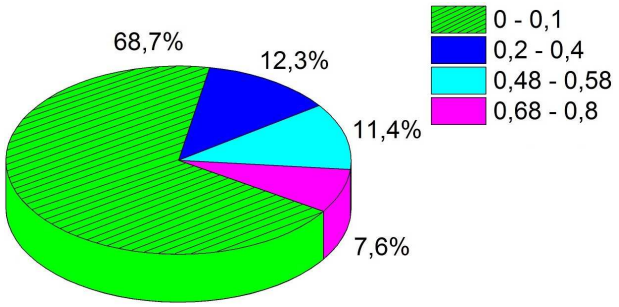


Figure 11: The percent of Speculative Queries with the estimated values of the Horizontal Selectivity for the restricted algorithm.

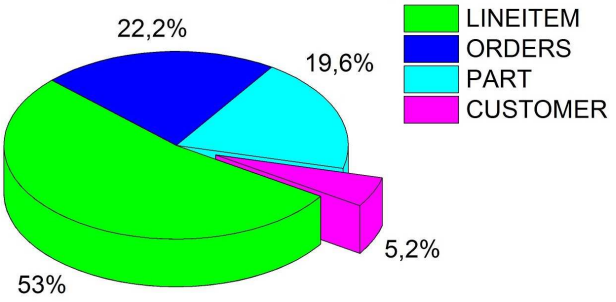


Figure 12: The percent of Speculative Queries executed for each of four database relations for the restricted algorithm.

means a two-sided value limitation. We can see that the two-sided limitation provides a fair Horizontal Selectivity and an average execution time reduction.

Table 3 presents the uniqueness of executed Speculative Queries. Value 90% for the EMB type of queries means that 10% of these queries has the structure identical to a previously executed (and removed from the Speculative DB) query. A special attention should be paid to the IN queries group, where there were no repeated queries. On the other hand, almost 60% of the LIKE queries had to be repeated due to the Speculative Analysis.

8.4 Multiple Speculative Queries with Restrictions

The extended algorithm for the execution of Speculative Queries excludes speculations with the Horizontal Selectivity > 0,8. Fig.11 presents the new distribution of the Horizontal Selectivity for Speculative Queries. We can see that now over 90 % of speculations have the Vertical Selectivity equal or less than 0,58. What's more, Fig.12 shows that a new group of queries appeared which is referring to the CUSTOMER relation. It has increased the use of multiple Speculative Query results in the input query execution.

Fig.13 shows how the Horizontal Selectivity restriction affected the percent of user queries which used the executed Speculative Query results. In comparison to Fig.6, we can see that for the template T3 we have managed to execute about 4 % of user queries using more than one Speculative Query. Additionally for templates T5, T7, T8 we have improved the number of user queries utilizing

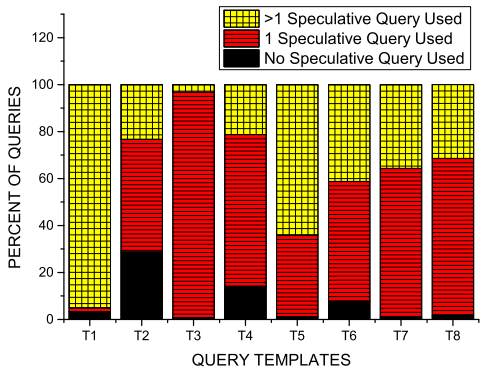


Figure 13: The percent of input queries which used the executed Speculative Query results for the restricted algorithm.

Speculative Queries, one or more, by 2 % on average. This improvement was possible due to a new group of speculations executed for the CUSTOMER relation. As a further consequence we have observed the average execution time reduction for templates T3, T5, T7, T8 from 0,5% to 1%.

9 CONCLUSION

The proposed middleware called the Speculative Layer is supporting SQL query execution in RDBMS. The presented model is based on joint graph modelling of group of queries. The generated Query Multigraphs are analysed to determine and execute different types of Speculative Queries. The proposed algorithm is aimed at the increased use of Speculative Queries in execution of input user queries. In this paper we have focused on improvement of multiple Speculative Query utilization by user queries. First series of experimental results for a test database and three sample sets of 1000 input queries are very promising. In the case of all 8 query templates more than 80% of input queries were executed with the use of at least one Speculative Query. A detailed analysis of obtained results has led us to important conclusions resulting in a modification of the execution algorithm for the Speculative Queries. The assumed further restrictions have eliminated the Speculative Queries execution with the almost full copy of the original database data (with the Horizontal Selectivity greater than 0,8). As a consequence, a new group of Speculative Queries appeared, providing further improvement in the use of multiple speculation results. On average 3% more user queries were executed with the use of more than one Speculative Query, providing further average query execution time reduction.

Further work will concentrate on more sophisticated methods used to determine the applied Speculative Queries. First, some effort should be put to avoid duplication of particular queries in favour of its higher diversification. In the context of the use of multiple Speculative Query results, the especially important seems to use Speculative Queries, which refer to diversified database relations.

Table 2: Average, maximal and minimal parameter values for each type of Speculative Queries.

		TYPE OF SPECULATIVE QUERY				
		IN	LIKE	EMB	*G||S*	*G&&S*
VERTICAL SEL.	MIN	0,66	0,61	0,59	0,71	0,61
	MAX	0,66	0,79	0,61	0,86	0,86
	AVG	0,66	0,69	0,6	0,72	0,75
HORIZONTAL SEL.	MIN	0,04	0,01	0,01	0,22	0,01
	MAX	0,1	0,2	0,03	1	0,98
	AVG	0,07	0,02	0,02	0,67	0,13
EXECUTION TIME	MIN	0,16	0,13	1,16	4,99	0,83
	MAX	0,38	1,56	1,27	12,18	9,45
	AVG	0,23	0,19	1,20	8,92	3,72

Table 3: The uniqueness of Speculative Queries i.e. the percent of queries executed only once.

	TYPE OF SPECULATIVE QUERY				
	IN	LIKE	EMB	*G||S*	*G&&S*
UNIQUENESS	100%	40,74%	90%	67,11%	84,4%

REFERENCES

[1] D. Kaeli, P. Yew, "Speculative Execution in High Performance Computer Architectures," Chapman Hall/CRC, 2005.
[2] A. Grama, A. Gupta, G. Karypis, V. Kumar, "Introduction to Parallel Computing (2nd Edition)," Addison Wesley, 2003.
[3] E.A.Jr. Liles, B. Wilner, "Branch prediction mechanism.," IBM Technical Disclosure Bulletin, 1979, Vol.22(7), p. 3013-3016.
[4] J.E. Smith, "A study of branch prediction strategies.," ISCA Conference Proceedings, 1981, New York, p.135-148.
[5] J. Puiggali, B. K. Szymanski, T. Jove, J. L. Marzo., "Dynamic branch speculation in a speculative parallelization architecture for computer clusters.", Concurrency and Computation: Practice and Experience, Vol.25(7), 2013, p.932-960.
[6] D. Padua, "Encyclopedia of Parallel Computing A-D.," Springer, 2011.
[7] N. Polyzotis, Y. Ioannidis, "Speculative query processing," CIDR Conference Proceedings, Asilomar, 2003, p.1-12.
[8] R.M. Karp, R.E. Miller, S. Winograd, "The Organization of Computations for Uniform Recurrence Equations," Journal of the ACM, 1967, Vol.14(3): p.563-590.
[9] G. Barish, C.A. Knoblock, "Speculative Plan Execution for Information Gathering," Artificial Intelligence, 2008, Vol.172(4-5), p.413-453.
[10] G. Barish, C.A. Knoblock, "Speculative Execution for Information Gathering Plans," AIPS Conf. Proceedings, Toulouse, 2002, p.184-193.
[11] V. Hristidis, Y. Papakonstantinou, "Algorithms and Applications for answering Ranked Queries using Ranked Views," VLDB Journal, 2004, Vol.13(1), p.49-70.
[12] A. Estebanez, D.R. Llanos, A. Gonzalez-Escribano, "A Survey on thread-Level Speculation Techniques.", ACM Computing Surveys, Vol. 49(2), 2016, p.22-39.
[13] A.Kejariwal, X.Tian, W.Li, M.Girkar, S.Kozukhov, H. Saito, U. Banerjee, A.Nicolau, A.V. Veidenbaum, C.D. Polychronopoulos, "On the performance potential of different types of speculative thread-level parallelism.", International Conference on Supercomputing Proceedings, 2006, Cairns, p.24.
[14] J. Šilc, T. Ungerer, B.Robi č, "Dynamic branch prediction and control speculation.", Int. Journal of High Performance Systems Arch., 2007, Vol. 1(1), p.2-13.
[15] S.T. Pan, K. So, J.T. Rahmeh, "Improving the accuracy of dynamic branch prediction using branch correlation.", International Conference on Architectural Support for Programming Languages and Operating Systems, 1992, Boston, p.76-84.
[16] A. Moshovos, S.E. Breach, T. N. Vijaykumar, G.S. Sohi, "Dynamic Speculation and Synchronization of Data Dependences.",24th ISCA, ACM SIGARCH Computer Architecture News, 1997, Vol.25(2).
[17] P.K. Reddy, M. Kitsuregawa, "Speculative locking Protocols to Improve Performance for Distributed Database Systems," IEEE Transactions on Knowledge and Data Engineering, 2004, Vol.16(2), p.154-169.
[18] T. Ragunathan, R.P. Krishna, "Performance Enhancement of Read-only Transactions Using Speculative Locking Protocol," IRISS, Hyderabad, 2007.
[19] T. Ragunathan T, R.P. Krishna, "Improving the performance of Read-only Transactions through Asynchronous Speculation," SpringSim Conference Proceedings, Ottawa, 2008, p.467-474.
[20] X.Ge, B.Yao, M.Guo, et al., "LSShare: an efficient multiple query optimization system in the cloud", Distrib. Parallel Databases, Vol.32(4), p. 593-605, 2014.
[21] M.B.Chaudhari, S.W.Dietrich, "Detecting common subexpressions for multiple query optimization over loosely-coupled heterogeneous data sources", Distrib. Parallel Databases, Vol.34, p.119-143, 2016.
[22] S.Y.Su, Y.Huang, N.Akaboshi, "Graph-Based Parallel Query Processing and Optimization Strategies for Object-Oriented Databases", Distributed and Parallel Databases, Vol.6(3), p. 247-285, 1998.
[23] G.Preti, M.Lissandrini, D.Mottin, Y.Velegrakis, "Mining patterns in graphs with multiple weights", Distributed and Parallel Databases, Special Issue on extending Database Technology, p.1-39, 2019.
[24] O.Goonetilleke, D.Koutra, K.Liao, T.Sellis, "On effective and efficient graph edge labeling", Distributed and Parallel Databases, Vol.37, p.5-38, 2019.
[25] H.M. Faisal, M.A. Tariq, Atta-ur-Rahman, A. Alghamdi, N. Alowain, "A Query Matching Approach for Object Relational Databases Over Semantic Cache", Chapter in Application of Decision Science in Business and Management, 2020.
[26] M. Ahmad, M. A. Qadir, M. Sanaullah, "Query Processing Over Relational Databases with Semantic Cache: A Survey", 2008 IEEE International Multitopic Conference, Karachi, 2008, pp. 558-564.
[27] F. Wang, G. Agrawal, "Query Reuse Based Query Planning for Searches over the Deep Web", Database and Expert Systems Applications. DEXA 2010. LNCS, Vol 6262, 2010.
[28] J.Gryz, "Query Optimization and Caching", Research Interests and Related Publications, Department of Computer Science York Univ., Toronto, Canada, 1998.
[29] P. Cybula, K. Subieta, "Query Optimization by Result Caching in the Stack-Based Approach", Objects and Databases. ICOODB 2010, LNCS, Vol.6348, 2010.
[30] A.Sasak-Okoń, Speculative query execution in Relational databases with Graph Modelling, Proceedings of the FEDCSIS, pp.1383-1387, ACSIS, Vol. 8., 2016.
[31] A.Sasak-Okoń, M.Tudruj, Graph-Based speculative Query Execution in Relational Data-bases, ISPDC 2017, July 2017, Innsbruck, Austria, CPS, IEEE Explore.
[32] A.Sasak-Okoń, M.Tudruj, Graph-Based speculative Query Execution for RDBMS, PPAM 2017, LNCS, vol 10777. Springer, Cham
[33] A.Sasak-Okoń, Modifying Queries Strategy for Graph-Based Speculative Query Execution for RDBMS, PPAM 2019, LNCS, Vol. 12043, pp. 408-418, 2020.
[34] TPC benchmarks, http://www.tpc.org/tpch/default.asp, 2015.
[35] G. Koutrika, A. Simitsis, Y. Ioannidis, "Conversational Databases: Explaining Structured Queries to Users", 2009, Tech. Report Stanford InfoLab.
[36] G. Koutrika, A. Simitsis, Y. Ioannidis, "Explaining Structured Queries in Natural Language.", ICDE Conference Proceedings, Long Beach, 2010, p. 333-344.

Hierarchical Embedding for DAG Reachability Queries

Giacomo Bergami
Dept. of Computing
Newcastle University, UK
ngb113@newcastle.ac.uk

Flavio Bertini
Dept. of CSE
University of Bologna, Italy
flavio.bertini2@unibo.it

Danilo Montesi
Dept. of CSE
University of Bologna, Italy
danilo.montesi@unibo.it

ABSTRACT

Current hierarchical embeddings are inaccurate in both reconstructing the original taxonomy and answering reachability queries over Direct Acyclic Graph. In this paper, we propose a new hierarchical embedding, the Euclidean Embedding (EE), that is correct by design due to its mathematical formulation and associated lemmas. Such embedding can be constructed during the visit of a taxonomy, thus making it faster to generate if compared to other learning-based embeddings. After proposing a novel set of metrics for determining the embedding accuracy with respect to the reachability queries, we compare our proposed embedding with state-of-the-art approaches using full trees from 3 to 1555 nodes and over a real-world Direct Acyclic Graph of 1170 nodes. The benchmark shows that EE outperforms our competitors in both accuracy and efficiency.

CCS CONCEPTS

• **Information systems** → *Storage architectures*; • **Theory of computation** → **Shortest paths**; *Sorting and searching*;

KEYWORDS

Hierarchical Embedding, Taxonomy, DAG, Tree, Reachability Query

ACM Reference format:
Giacomo Bergami, Flavio Bertini, and Danilo Montesi. 2020. Hierarchical Embedding for DAG Reachability Queries. In *Proceedings of 24th International Database Engineering & Applications Symposium, Seoul, Republic of Korea, August 12–14, 2020 (IDEAS 2020),* 10 pages.
https://doi.org/10.1145/3410566.3410583

1 INTRODUCTION

Hierarchical information allows a compact and unambiguous representation of *is-a* relationships (i.e., edges) among entities (i.e., nodes). Hierarchical information can be represented either as a Direct Acyclic Graph (DAG) [11] or as a tree [15]: while the former allows multiple parents per node, such as entity d in Figure 1a, the latter requires each node to have at most one parent (Figure 1b). Therefore, the most general representation of hierarchical information is via DAG with a *root* node. We refer to this specific type of DAG as a Hierarchical DAG. Both Hierarchical DAGs and Trees are rooted in one entity (represented as the *root*), describing the maximum degree of generalization. The *root* node both has no ancestors and is the common ancestor of all the other nodes. By transitive closure, paths in Π_n from any entity n towards the *root* represent all of its possible generalizations, namely **generalization path** [16].

The Hierarchical DAG in Figure 1 represents a toy example of *is-a* relationships available from WordNet [14]: a *Dog* and a *Bee* are both *Domestic Animals*, but only the former is a *Mammal*; the *Zebra* is a *Mammal*, and therefore an *Animal*, which was never domesticated. Such data structure also induces a notion of similarity among the represented nodes: a *Dog* will be more similar to a *Mammal* than any other *Animal*, while a *Dog* has little to share with a *Bee*.

Hierarchical information is also commonly used in Data Warehouses [2] for enabling aggregations (*roll-up*) and disaggregations (*drill-down*) over categorical multidimensional data. Each data warehouse can be represented as a *data (hyper)cube*, where each possible combination of dimensions' values identifies a fact, which is usually associated to a numerical measure. Figure 2a represents a data cube for a Sales relation, where Products, Stores, and Dates are the dimensions of reference, each fact (represented as a small cube) represents the number of sales, and each dimension is associated to a hierarchy. In particular, Dates' possible values can be represented as a DAG: each day can be aggregated either by week (*Week 4*) or by month (*January*), but weeks cannot be aggregated as months or viceversa; last, months and weeks can be aggregated into years (*2020*). Stores aggregation uses geographic data, which can be also intuitively represented as Hierarchical Trees[1]: e.g., all the shops in *Tokyo* belong to the *Japan* division, which is a part of all the shops in *Asia* and then part of the organization as a whole (*World*). By exploiting the generalizations induced by each hierarchy, we can also produce the rolled-up data (hyper)cube (Figure 2b). Given that all the possible fact generalizations can be represnted as a DAG (Figure 2c) we can exploit DAG embeddings to potentially enhance multidimensional (graph) pattern mining requiring generalization paths [16].

In current literature, we refer as *embedding* to any (compact) vectorial representation of graph or tree data which, via morphisms, boil hierarchical operations down to vector operations; given that hierarchical data usually provide one single type of relationship[2], the embedding only focuses on providing a vectorial representation of the nodes. Figure 1c provides an possible way for the embedding for the tree in Figure 1b via our proposed embedding: the vectorial representation empowers the definition of a vector distance as a value proportional to the actual distance between the nodes in the Hierarchical DAG and Tree. In other words, the more the nodes are distant within the hierarchy path, the less will be the shared

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410583

[1] http://www.geonames.org
[2] E.g., [15] ignores synonymy, relatedness, and antimony, but only focuses on *is-a* relationships

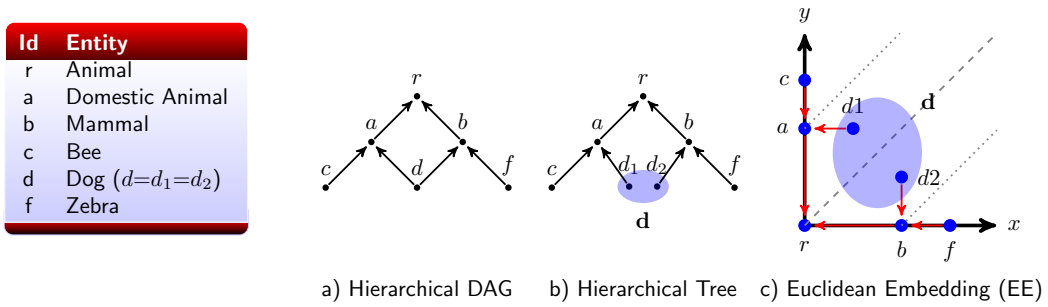
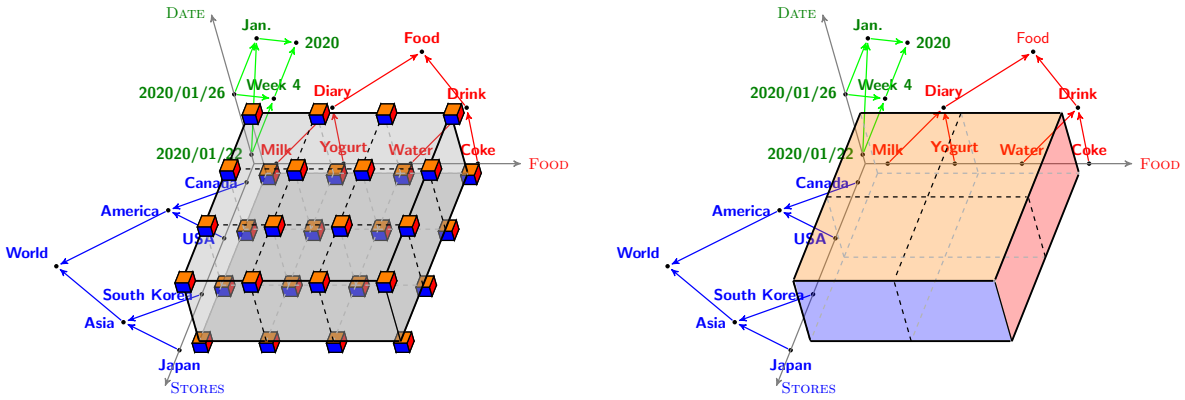
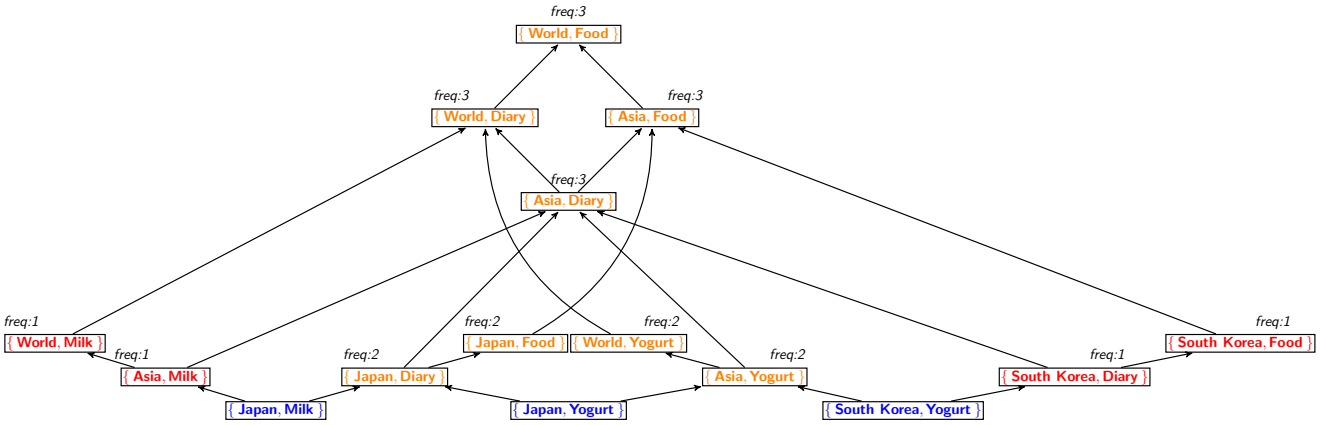


Figure 1: Given a generic DAG rooted in r (a), we can always represent it as one single hierarchical tree (b) for which the distance between the siblings is always greater than the one to the parent (c).



(a) Unaggregated and original Data Warehouse. **(b) Data Warehouse rolled up by continent, year and food kind.**



(c) Representing all the possible generalizations (with their associated frequencies) of three facts (in blue) from the data warehouse over the Food and Stores dimensions. The generalizations shared among the facts are coloured in orange.

Figure 2: A three-dimensional Data Warehouse (a) can be expressed as a lattice (c) enabling a compact representation for frequent pattern mining and similarity queries via its associated embedding.

similarity. Given the similarity-distance duality [4], we improperly refer to those as **metrics** in the rest of the paper. The vectors' distances can be used to reconstruct the original node hierarchy and therefore, vectors can be used for node reachability queries, that can be computed in constant time with a distance metric of choice. These considerations are also beneficial to database inconsistency

Table 1: Converting a path represented as in [16] to the proposed Euclidean Embeddings: δ represents the *distance factor* while λ represents the *decay factor* exponentially increasing with the distance from the root. From n ="dog" we have $2 = |\Pi_n|$: we generate two vectors for each path in Π_n.

n	Name	Gen. Paths (Π_n)	EE $\mathbb{R}^b$ for $b = 2$	Height (h, k)
r	Animal	$\emptyset$	$\{ (0, 0) \}$	0
a	Domestic Animal	2	$\{ (0, \delta) \}$	1
b	Mammal	1	$\{ (\delta, 0) \}$	1
c	Bee	2.2	$\{ (0, \delta + {}^{\delta}/_{\lambda}) \}$	2
d	Dog	2.1, 1.2	$\{ ({}^{\delta}/_{\lambda}, \delta),\ (\delta, {}^{\delta}/_{\lambda}) \}$	2
f	Zebra	1.1	$\{ (\delta + {}^{\delta}/_{\lambda}, 0) \}$	2

detection algorithms [1, 6]. With reference to the previous Data Warehouse example, we can then represent the DAG in Figure 2c via embeddings. We might then exploit vectorial indexing techniques for multidimensional data [3] for fastening top-k similarity queries based on the hierarchical information. Last, given that frequent pattern mining techniques often include hierarchies for generalizing frequent patterns (Figure 2c), such embeddings might also speed up frequent pattern matching algorithms such as [16].

At the time of the writing, the most qualitatively performant entity embedding for hierarchical data [15] exploits a **Poincarré Hyperbolic Disc**. A Poincarré Hyperbolic Disc (Figure 3) is a hyperbolic geometry where all the points are within the unit disk, i.e. $P^b = \{\vec{x} \in \mathbb{R}^b \mid \|\vec{x}\|_2 = 1\}$, and the straight lines mainly consist of all circular arcs. In particular, distances are defined via *Cayley–Klein metrics*: each branching in the tree in Figure 3 is defined to have the same length [15], so that the more that we approach the border of the disk, the more the distances will increase. The proposed embedding on the Poincarré Hyperbolic Disc has several limits. First, hierarchies with a high branching factor represent the worst-case scenario for vector embeddings: leaves at maximum tree depth can both be packed near the ball's boundary and potentially nearer to each other than any parent node. The reconstruction of the original hierarchical data structure via such representation is often inaccurate. Last, the preliminary training required to generate the vector embedding causes two other problems: *(i)* given that the learning model might provide a non-zero loss, the reconstructed hierarchy from such embeddings will not precisely match the original representation, and *(ii)* the computational cost of the training phase is greater than a simple hierarchy visit, and so we cannot adopt such a technique for indexing big data as Data Warehouses.

In this paper, we propose a novel hierarchical embedding for multidimensional data indexing, Euclidean Embedding[3] (EE), arising from these limitations: we might not generate such vectors via learning, and we require that the vector ranking strictly meets the requirements stated in the former paragraphs. With respect to the ranking, given a node u, the first ranked k elements should be the nearest k ancestors or descendants of the node u sharing a same generalization path. All the unreachable nodes should be ranked with the highest possible value, e.g. $+\infty$. Our proposed embedding associates to each node multiple vectors in the Euclidean space, where each vector's dimension is bounded by the maximum branching factor of the hierarchy, and the number of vectors for each node reflects the number of its generalization paths. Figure 1c

[3]Dataset and surce code: https://github.com/gyankos/hierarchy_embedding

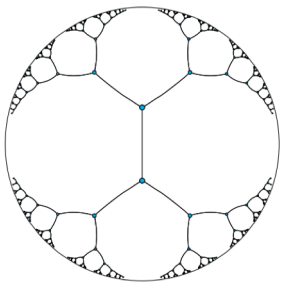


Figure 3: Hierarchy in a 2D Poincarré Hyperbolic Disc [15].

jointly with Table 1 provide an intuition on how to generate the EE from a path traversal of a Hierarchical Tree (Figure 1b): while moving from a parent node n to the i-th child c at depth level j, we add $\frac{\delta}{\lambda^{j-1}}$ to $\mathbf{v}_n$ for obtaining the new embedding $\mathbf{v}_c$. Please note that this result does not contradict the recently proved inability of representing graphs in low-dimensional Euclidean spaces [18], as we show that one possible alternative for keeping the dimensionality low for each vector is to generate multiple possible vectors. This approach could be also generalized for rooted DAGs (Figure 1a), by replacing each node n in the DAG with as many nodes as all the possible generalization paths $|\Pi_n|$ connecting it to the root, and by connecting each of these demultiplexed nodes to only one of the original parents of n. As a result, the final DAG entity n will be associated with as many vectors as the generalization paths (see §3). This solution is practical in real world hierarchies, as the great majority of the nodes has only one ancestor [14] (see §5.2).

The proposed EE is also going to validate the following claims:

- it is possible to represent each entity node n of a hierarchical tree having a maximum branching factor of b as a b-dimensional vector $\mathbf{v}_n$ in the Euclidean space ($\mathbb{R}^b$),
- it is possible to define a distance metric over these vectors so to distinguish the elements that share a generalization path from those that do not, and
- there is a mapping $\mathcal{G}$ from rooted DAGs to transformed Trees, thus associating one single DAG entity to one or more vectors in EE generated over such Trees.

After defining our proposed EE embedding (§3), we show that we can determine with absolute precision the elements sharing a generalization path via an associated metric (§4). We motivate such assertion via both an intuition of the proof and some empirical experiments. Those are going to use novel **accuracy measures** for determining the embedding and their associated metric's precision in reconstructing the taxonomy structure (§5). We are also going to show that EE and its metric can be computed faster than the one from the most accurate competitor. This claim is supported through benchmarks over different competing hierarchical embedding requiring either learning [10, 15] or just a preliminary visit of the data structure [13, 21] as our proposed EE. Such benchmarks are performed over both Hierarchical Trees (§5.1) and Hierarchical DAGs (§5.2). We draw our conclusions and propose some future works in §6.

Table 2: Related Work sorted by descendant generation time computational complexity. κ denotes a predefined embedding size, G is a shorthand for the hierarchy size in $O(|V| + |E|)$, b and h are respectively the taxonomy's maximum branching and maximum height, e is the number of epochs and m is the minibatch size.

	Generation Approach	Space for n	Generation Time Bound	Metric Type	$d(u, v)$
E [10]	Euclidean Learning	κ	$O(\lvert V\rvert emb^h(b^h\kappa + \lvert V\rvert^2))$	distance	$(u-v)^t M_{u,v}(u-v)$
P [15]	Poincarré Learning	κ	$O(\lvert V\rvert(2\lvert V\rvert + em))$ [12]	distance	$\operatorname{acosh}\left(1 + 2\frac{\lVert u-v\rVert^2}{(1-\lVert u\rVert^2)(1-\lVert v\rVert^2)}\right)$
PHC [21]	Visit, Matrix SVD	κ	$O(G \cdot \kappa^2 \lvert V\rvert)$	distance	$\lVert u-v\rVert^2$
RV/LD/MD* [13]	Visit, Node density	$\lvert V\rvert$	$O(G \cdot \lvert V\rvert)$	similarity	$\cos(u, v)$
PHS [21]	Visit, Average	κ	$O(G \cdot b)$	distance	$\lVert u-v\rVert^2$
EE (*this paper*)	Visit	$O(b \cdot \lvert\Pi_n\rvert)$	$O(G \cdot b)$	distance	*Definition 4.3 on page 6*

2 RELATED WORK

Hierarchical Embeddings. Table 2 summarizes the most relevant similarities and differences among different state of the art approaches. We now discuss those in detail.

One of the first vector embeddings for representing entities in Hierarchical Trees without using any additional corpus information is [13]. This feature is extremely useful for settings where only *folksonomies* are possible [9], where new taxonomical terms might be introduced for the first time by non-experts. After enumerating the entities following a level order traversal, they associate to each entity n a density value δ_n that can be calculated by using different strategies: *Relevance Vector* (**RV**), *Local Density* (**LD**) and *Multiple Descent* (**MD**). Then, each vector embedding $\mathbf{v}_n$ has a non-zero density value $\delta_{n'}$ as its n'-th vectorial component if and only if entity n' is either an ancestor or a descendant of n. The relevance of different density δ_n can be fine tuned using some parameters (α and β). This approach has two shortcomings: first, the embedding size increases with the number of the nodes, thus making it impractical to fit all the vector representations in main memory for big data scenarios; second, the similarity score of the embedding reflects more the density and the path distance between the nodes than their actual distance within the generalization path. Given that the authors provided no implementation of their proposed embedding, we implemented it from scratch in our codebase.

Entity Embeddings (**E**) [10] followed a completely different approach to represent hierarchies, as the authors represent correlation or synonym relationships by associating multiple entities with a single category in the hierarchy. As opposed to the previous model, their data model supports Hierarchical DAGs. Instead of learning an explicit vector representation for each node in the hierarchy, they learn vectors for each entity contained in a category, which is then represented as a matrix. When those are combined in a metric, they provide a metric distance between categories defined by chaining the vectors and matrices $M_{u,v}$ within the path $u \rightsquigarrow v$ of the extended Hierarchical DAG; the authors also provide a top-k entity similarity query. Both the proposed data structure and the query concept introduced by the authors are more general than the one proposed in the present paper, and therefore we can use E in our experiments. This approach has two major shortcomings: first, this approach requires to learn not only vectors for each entity but also a matrix for each category, thus increasing the data volume required by the final metric; second, as for any other learning-based approach, the training time increases with the embedding dimension κ and with the size of the dataset itself, and it becomes quickly impractical for taxonomies containing at least 1500 nodes. Given that the authors provide a C++ implementation of both the training and the query algorithm as well as a top-k search, we import a working subset of their repository[4] in our codebase.

Par2Hier [21] generates a hierarchical tree representation of a text corpus divided into paragraphs, sections, and documents. Corpora hierarchical embeddings are generated from the vector representation (e.g., via *Word2Vec*) of the textual entities contained in each paragraph. This represents the major shortcoming of Par2Hier because such embedding might not always be available as per previous discussions. According to the author's intentions, the ultimate goal of the paper is to create hierarchical embeddings for a corpus considering the relationships between entities in a hierarchy. The author follows two separate approaches to generate the parent vector from the child vectors: either the parent vector is a sum of the child vectors (**PHS**) or it is the outcome of a truncated Singular Value Decomposition of the matrix, which columns are the child nodes' vectors (**PHC**). Accuracy measures used in benchmarks are limited to the likelihood that a vector $\mathbf{v}$, neighbour of a paragraph vector belonging to a document d, belongs to d itself. This approach, entirely implemented in Java[5], has been completely rewritten in C++ using Eigen3 [7] for linear algebra operations.

Last, Poincarré Embeddings[6] (**P**) [15] provides vector embeddings for nodes in both DAG and Tree Hierarchies. The authors claim that, in an Euclidean Space, complex graph patterns require a computationally infeasible embedding dimensionality to accurately represent hierarchical embeddings without loss of information. Therefore, they provide a representation of such tree using an n-dimensional Poincarré Hyperbolic Disk space. n-dimensional vectors are learned in this space by using a transitive closure of all the edges in the hierarchical dataset. Then, the vector distance in the Poincarré Hyperbolic Disk space is expressed in terms of acosh and Euclidean Norm $\lVert\cdot\rVert_2$. Albeit they used mean average precision and some ranking as their accuracy measures, in the former they did not restrict the precision to the first $k+1$ distant elements where k is the maximum length of a generalization path, and in the latter

[4] https://github.com/ZhitingHu/EEEL
[5] https://github.com/tteophile/par2hier
[6] https://github.com/facebookresearch/poincare-embeddings

they did not provide a normalized ranking comparison as the Spearman Correlation, thus making it quite hard to compare different ranking results over different embedding size and datasets. Given that the authors' assumptions over a DAG and Tree representation of hierarchical data are the same as the one provided in the present paper, we directly train their model and use the generated node embeddings in our testing framework.

Distance Geometry Problem. Distance Geometry Problem (DGP) is the mathematical formulation for the generic embedding problem in the Euclidean Space for graphs: given a weighted graph and a embedding space size b, we want to associate to each node in the graph a b-dimensional vector such that their associated vector distances are the same as the nodes' distances in the graph. This problem guarantees that the distance between embeddings of adjacent nodes matches the graph distance, but does not guarantee that the distance between two arbitrary vectors is proportional to their distance within the graph. Albeit it is very well known that the exact solution for this generic problem is NP-Hard [17], the approximate solution has a tractable solution for low dimensions [22]. Furthermore, recent research showed that current approaches in representing graphs via low-dimensional embeddings fail to properly solve the well known DGP [18]. In this paper we find a exact and feasable solution for a specific sub-problem of the DGP: we consider rooted unweighted DAGs instead of weighted graphs but, instead of returning one single vector, we allow that one node is associated to more than one vector.

DAG Exact Reachability Queries. Given a graph G, the reachability query $u \rightsquigarrow v$ over two nodes u and v returns true if and only if it exists a path in G connecting u to v. Methods for reachability queries in acyclic graphs in constant time cannot be used to generate embeddings [23]. As an example, dual labelling [8] or graph transitive closing approaches [19] do not generate embeddings and, therefore, cannot be used in conjunction with other semantic techniques requiring a metric between two (connected) concepts. Also, techniques allowing efficient query time do not provide an adequate indexing time (e.g., the search takes place in constant time, while the indexing is quadratic in the size of the graph nodes). On the other hand, the approach we have proposed provides an good indexing time, and still compatible with the visiting time of the graph data structure, $O(|G|) = O(|V| + |E|)$. Given that trees are a specific case of DAGs, such general results can also be applied to hierarchical trees.

3 EUCLIDEAN EMBEDDING

First, since hierarchies can be expressed both as trees and as DAGs, we extend the notion of taxonomy in [16], as it is important in understanding the reasons behind the proposed Euclidean Embedding.

Definition 3.1 (Taxonomy). A *taxonomy* could be represented by either a Hierarchical DAG or as a Hierarchical Tree. A **Hierarchical Graph** is a triple $G = \langle N_G, R_G, \ell_G \rangle$ representing a Direct Acyclic Graph rooted in ℓ_G: the entities are represented as a set of nodes N_G represented by string labels, and R_G is the set of the is-a relationships. The root node $\ell_G \in N_G$ has no parent nodes. A **Hierarchical Tree** T is a particular Hierarchical DAG where all the entity nodes in $N_T \backslash \{\ell_T\}$ have only one parent. Each **generalization path** from a node n in N_T (N_G) to the root ℓ_T (ℓ_G) is represented by a list of non-zero integers $c_1, \ldots, c_m$, where n is the c_m-th child of his parent and the second-last ancestor of n is the c_1-th child of the root. The root will have an empty generalization path. Each node of the taxonomy is uniquely identified by the set of all the possible generalization paths generated from it. □

Algorithm 1 Generating a Hierarchical Tree T with mapping $\mathcal{G}$ from a Hierarchical DAG G.

```
1:  procedure GenerateEmbeddings(G = ⟨N_G, R_G, ℓ_G⟩)
2:      global 𝒢(ℓ_G) := {ℓ_G}; T := ⟨N_T = ∅, R_T = ∅, ℓ_G⟩ ; b = −∞
3:      DFSVisit(G, 0, ℓ_G)
4:  procedure DFSVisit(G, h, n, l)
5:      b := max{b, | ↑_n |}; isTree:=| ↑_n | ≤ 1
6:      for each parent p_i^n of n do                          ▷ p_i^n ∈ ↑_n
7:          for each p̃ ∈ 𝒢(p_i^n) do    ▷ If p_i^n is associated to nodes in T
8:              n'_i := isTree ? n : fresh_id()
9:              N_T := N_T ∪ {n'_i}; R_T := R_T ∪ {n'_i → p̃};
10:             𝒢(n) := 𝒢(n) ∪ {n'_i}
11:     for each descendant d of n do DFSVisit(G, h + 1, d)
```

Each node n in a Hierarchical Tree[7] T will be associated to just one single generalization path π_n, which can be trivially generated by visiting the R_T edges backwards from the root. For each of these paths, we can generate the associated EE as follows: given b the maximum branching factor of a tree T, the node ℓ_T will be placed at the origin of the axes, i.e. $\mathbf{v}_{\ell_T} = \vec{0} \in \mathbb{R}^b$. Each j-th child m of any node n for which the embedding $\mathbf{v}_n$ is already known, the embedding of m is $\mathbf{v}_m = \mathbf{v}_n + (\underbrace{0, \ldots, 0}_{j}, \frac{\delta}{\lambda^{h-1}}, \ldots, 0) \in \mathbb{R}^b$, where h is the distance of m to the root (*height*). The application of such procedure to the Hierarchical Tree in Figure 1b generates the EE embeddings in Figure 1c.

On the other hand, each node n in a Hierarchical DAG G will be associated to more than just one generalization path $\pi_{n'}$. In order to ease the embedding definition for the Hierarchical DAG G, we can first transform it into a Hierarchical Tree T, so that a DAG node n is associated to multiple Tree nodes n' having one single generalization path; then, we generate the embedding for n as a set of vectors generated from the associated n'-s in T. With reference to the introductory mammal toy dataset, Figure 1 shows the outcome of the transformation of G into T, while Table 1 provides the resulting embedding for G.

Algorithm 1 transforms the Hierarchical DAG into a Hierarchical Tree: while visiting the DAG using a depth first search from the root (Line 3), we can gradually transform it into a tree T by replacing each currently visited node n with as many nodes $\mathcal{G}(n) = \{n'_1, \ldots, n'_b\}$ as the number of all the generalization paths originated from the parents in $\uparrow_n = \{p_1^n, \ldots, p_b^n\}$ (Line 7). Next, we connect each node n'_i (Line 10) with an edge $n'_i \rightarrow \tilde{p}$ to one of correspondent tree parents $\tilde{p}$ (Line 9) and, by recursively visiting the graph (Line 11), to all the descendants of n so that, after completely visiting the graph, we will obtain the Hierarchical Tree in Figure 1: the $\mathcal{G}$ function

[7] In the following statements, we use the same notation from Algorithm 1, which converts a Hierarchical DAG into a Hierarchical Tree via a DFS visit.

maps each node in the original Hierarchical DAG to the ones in the Hierarchical Tree. After running such algorithm and obtaining a morphism $\mathcal{G}$ and a Hierarchical Tree T, each node n in such a DAG is associated to a set of vectors $\{\mathbf{v}_{n'} | n' \in \mathcal{G}(n)\}$, while each Hierarchical Tree node will be associated to one single vector.

We can provide a formal definition of the embedding from the following formal definition of an Euclidean Embedding subsuming the transformation from the previous algorithm:

Definition 3.2 (Euclidean Embedding, EE). Given a node n within a taxonomy having maximum branching factor b, its Euclidean Embedding is defined as a set of vectors, having a vector $\mathbf{v}_{n'} := \left(\delta \cdot \Sigma_{\substack{1 \leq i \leq h \\ c_i = j}} \lambda^{-(i-1)}\right)_{1 \leq j \leq b}$ for each **generalization path** $c_1, \ldots, c_h$ in Π_n associated to n.

In particular, $\delta \in \mathbb{N}$ is the *distance factor* representing the distance between the root and its descendants, and $\lambda > 1 \in \mathbb{N}$ is the *decay factor* such that the distance of a node from its ancestors increases with an inverse exponential factor λ^{-i}. □

Example 3.3. Using Figure 1 as a running example, Table 1 provides all the EE generated for each node n by exploiting the generalization paths in Π_n.

Given that computers represent decimal numbers in basis 2, we use $\lambda = 2$ for making EE numerically robust, and we chose $\delta = 3$ to minimize the roundoff errors. Please observe that, given the specific choice of $\lambda = 2$, it is also very simple to infer the height of a node n: by the former definition, the height of a node will be associated to the minimum non-zero bit in the mantissa of each component's double of the vector $\frac{\mathbf{v}_n}{\delta\lambda}$.

4 EUCLIDEAN EMBEDDING'S DISTANCE METRIC

At this point, we possess all the hints to determine the desired metric for the proposed EE: we can observe that, given a generic non-leaf node n within a Hierarchical Tree with a generalization path $c_1, \ldots, c_n$, his further ancestors at distance k will be the ones having generalization paths $c_1, \ldots, c_n, c_{n+1}, \ldots, c_{n+k}$ where $c_{n+1} = c_{n+2} = \cdots = c_{n+k}$. Using a Hierarchy Tree with maximum branching factor $b = 2$ as an example, the furthest node from the root will be its leftmost and its rightmost descendants (Figure 1). Following similar considerations, we might observe that the maximum distance between two nodes n and m sharing a same generalization path and respectively at height h and k with $h \geq k$ is $\delta \sum_{i=k}^{h-1} \lambda^{-i}$. From this intuition, we derive the following upper bound:

Lemma 4.1. *Given two nodes n and m within a Hierarchical Tree respectively at height h and k with $h \geq k \geq 0$, we state that m appears in the generalization path of n if and only if the following condition holds:*

$$\|\mathbf{v}_n - \mathbf{v}_m\|_2 \leq \delta \sum_{i=k}^{h} \lambda^{-i}$$

We will define such upper bound as the following threshold shorthand: $\theta(h,k) := \delta \sum_{i=k}^{h} \lambda^{-i}$

The formal proof by contradiction is omitted due to the lack of space, but we can now provide an intuition for it. Let us consider node b and c from Figure 1: they are respectively at height 1 and 2, but they do not share a generalization path. We want to assess whether threshold θ will still hold in this case: we will have that $\|\mathbf{v}_b - \mathbf{v}_c\|_2 = \sqrt{2\delta^2 + \frac{\delta^2}{\lambda^2} + 2\frac{\delta^2}{\lambda}} \stackrel{?}{\leq} \frac{\delta}{\lambda} + \frac{\delta}{\lambda^2}$, which is never verified. As also revealed by some empirical stress tests[8], this happens only when two nodes do not share a generalization path.

Example 4.2. Matrix $\Theta_{h,k} := \theta(h,k)$ represents all the possible threshold values for all the possible heights in Figure 1 for $\delta = 3$ and $\lambda = 2$:

$$\Theta := \begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 0 & 1 & 2 \end{array} \\ \begin{bmatrix} 3.00 & 4.50 & 5.25 \\ 4.50 & 1.50 & 2.25 \\ 5.25 & 2.25 & 0.75 \end{bmatrix} \end{array}$$

Now, we can use this given threshold to separate all the nodes that should lie in a generalization path from the ones that do not, by associating an infinite distance to the latter ones. Even in this case, we are going to provide an informal proof for this observation: from the former lemma, all the nodes satisfying the upper bound will be all nodes sharing a generalization path and located at a distance $h - k$. If we now consider all the nodes m' at a height $k - 1$ from the root, then these nodes will be located at a greater upper-bound distance from n, i.e. $\theta(h, k-1)$. Given that by construction $\theta(h, k-1) < \theta(h,k)$ and given that θ is providing an upper-bound to the metric's distances, then we will be always able to ascertain the ranking position within the generalization path by ordering the nodes by increasing distance. We can now define the following metric:

Definition 4.3 (EE Metric). Given two nodes n and m within a Hierarchical Tree respectively at height h and k with $h \geq k \geq 0$, their associated distance is computed as follows:

$$d_T(n,m) = \begin{cases} \|\mathbf{v}_n - \mathbf{v}_m\|_2 & \|\mathbf{v}_n - \mathbf{v}_m\|_2 \leq \theta(h,k) \\ +\infty & \text{oth.} \end{cases}$$

where $+\infty$ represents the maximum value representable as a IEEE 754 double, that is returned only if the nodes do not satisfy the θ upper bound, and therefore are predicted not to lie a common generalization path. This metric can be also generalized to the Hierarchical Graphs as follows:

$$d_G(n,m) = \min_{n' \in \mathcal{G}(n), m' \in \mathcal{G}(m)} d_T(n', m')$$

Finally, we can define the **similarity function** between two Taxonomy nodes after normalizing their distance [4] as follows:

$$\text{sim}^d(n,m) = 1 - \frac{d(n,m)}{1 + d(n,m)}$$

where either $d = d_T$ or $d = d_G$. □

Even in this case, we can informally prove that the d_T metric is a proper distance metric (i.e., it satisfies the triangular inequality) and that d_G returns either a value which is proportional to the length of the shortest path between the two nodes or $+\infty$. In fact, if there exists a minimum path between n and m in a DAG, then the distance metric will return the minimum distance between the nodes via the ranking induced by the upper-bound provided

[8] https://gist.github.com/jackbergus/0e944b1b62155a655cc4d97df9b09d6c

by Lemma 4.1, and therefore it will represent the distances of the nodes within such minimum path. All the pairs of nodes that do not share a generalization path will not be reachable by definition, and therefore their distance should be necessarily $+\infty$. Given, per previous considerations, that the threshold θ is exceeded only when the nodes do not share a generalization path and therefore $+\infty$ is returned, then d_T and d_G reflect the expected notion of distance within a taxonomy.

Example 4.4. Matrix $M_{uv} := d_G(u, v)$ represents the distances calculated from d_G over the embeddings associated to the entities in Table 1 with $\delta = 3$ and $\lambda = 2$. As we would expect, $M_{uv} = +\infty$ if and only if the Euclidean distance between u and v is above the threshold θ, thus implying that those nodes do not share a generalization path:

$$M := \begin{array}{c} \\ r \\ a \\ b \\ c \\ d \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} r & a & b & c & d & f \end{array} \\ \begin{bmatrix} 0 & 3.00 & 3.00 & 4.50 & 3.35 & 4.50 \\ 3.00 & 0 & +\infty & 1.50 & 1.50 & +\infty \\ 3.00 & +\infty & 0 & +\infty & 1.50 & 1.50 \\ 4.50 & 1.50 & +\infty & 0 & +\infty & +\infty \\ 3.35 & 1.50 & 1.50 & +\infty & 0 & +\infty \\ 4.50 & +\infty & 1.50 & +\infty & +\infty & 0 \end{bmatrix} \end{array}$$

5 BENCHMARKS

Algorithm 2 provides the pseudocode for the benchmark used to compare the Euclidean Embedding (§3) and our proposed metric (§4) with the other competitors. This pseudocode also applies for all types of taxonomic data. We now outline for the first time – to the best of our knowledge – some **accuracy measures** for precisely determining whether an embedding with an associated metric is able to correctly separate the elements belonging to a same generalization path from the ones that do not. In order to meet our goal, we used *Precision@≤k*, *Recall@>k*, and the *Spearman correlation index*. For all the generalization paths Π_n from a leaf node n with length k providing us the ground truth, for each embedding strategy $\mathbf{v}$ and associated vector distance d, we compute the node embedding $\mathbf{v}_x$ of each node $x \in N_G$ in the hierarchical data structure and rank it with natural numbers $i \in \mathbb{N}$ by increasing distance with $\mathbf{v}_n$ (Line 8): we expect that the values with at most rank k will be predicted members of the generalization path of n (Line 11), and the remaining ones should be nodes not related to n (Line 13).

Precision@≤k checks how many elements that have a rank of at most k are actually within the same hierarchical path (Line 29): an accurate metric over a specific embedding should always return 1. *Recall@>k* is used to determine if there are any false negatives; in particular, it checks the correctness of a metric over a specific embedding by verifying whether any node with a rank greater than k does actually belong to a hierarchical path (Line 30): a precise metric should always return 0, because none of these nodes should belong to the generalization path. *Spearman correlation index* measures the normalized variance between the ranking induced by a metric over a specific embedding and the expected ranking induced by the generalization path (Line 32). Therefore, this accuracy measure will determine if the metrics and their associated embedding accurately reconstruct the original taxonomy with respect to the reachability query. We also impose that all the nodes not in a generalization path should be all ranked as in the $(k + 1)$-th position, so we can always discriminate between the nodes in the generalization paths from the ones who aren't when the value of k is unknown (Line 26). A precise metric should always return 1 (*positive correlation*), while values near zero mean that there is almost no correlation between the returned and the expected ranking (*null correlation*). Negative values of the metric provide an inverse ranking correlation (*negative correlation*).

Algorithm 2 Evaluating the Hierarchical Embeddings $\mathbf{v}$ with their associated distance metrics d over a hierarchy G by generating a generalization path from n. The accuracy measures are benchmark configuration independent (Lines 29-32).

```
1:  global rankedMap={}, predictedPositive= ∅,
2:  global predictedNegative= ∅, avgRank=0;
3:
4:  procedure NormalizeMapRank(N_G, v, d, n, i)
5:      if N_G ≠ ∅ then
6:          r := min_{m∈N_G} d(n, m)
7:          for each m ∈ N_G s.t. d(n, m) = r do
8:              rankedMap[m]:=i; N_G := N_G\{m}
9:              avgRank+=i
10:             if i ≤ k then
11:                 predictedPositive:= predictedPositive∪{m}
12:             else
13:                 predictedNegative:= predictedNegativeU{m}
14:         NormalizeMapRank(N_G, v, d, n, i + 1)
15:
16: procedure RankWithDistanceMetric(G = ⟨N_G, R_G, ℓ_G⟩, v, d, n)
17:     expectedMap = {};      P = {n}
18:     expectedMap[n] = 0
19:     for each π = n → m → o ... z → ℓ_G ∈ Π_n do
20:         P := P ∪ {m, o, ..., z, ℓ_G}                ▷ Positive Examples
21:     for each x ∈ P do
22:         expectedMap[x]:=min_{p∈n⇝v} |p|              ▷ Expected Ranking
23:     N := N_G\P                                      ▷ Negative examples
24:     k := max_y expectedMap[y]
25:     avgExpected= (Σ_{i≤k} i+|N|(k+1)) / |N_G|
26:     for each n ∈ N do expectedMap[n]=k + 1
27:     NormalizeMapRank(N_G, d, v, n, 0)               ▷ Metric-induced ranking
28:     avgRank= avgRank / |N_G|
29:     precision@≤ k = |predictedPositive∩P| / |P|
30:     recall@> k = |predictedNegative∩P| / |predictedNegative|
31:     Spearman=
32:     Σ_{i∈N_G}(rankedMap[i]−avgRank)(expectedMap[i]−avgExpected) / (Σ_{i∈N_G}(rankedMap[i]−avgRank) Σ_{i∈N_G}(expectedMap[i]−avgExpected))
```

Given that all these accuracy measures are normalized, we can compute their values for each of the generalization paths from the leaf nodes and then do their average. As a result, we can now compare different averaged results over disparate dataset sizes. We performed our tests described in the following subsections over a Lenovo ThinkPad P51 with a 3.00 GHz (until 4.00 GHz) Intel Xeon processor and 64 GB of RAM at 2.400 MHz.

5.1 Hierarchical Trees

Dataset. The tree dataset was procedurally generated for stress testing the accuracy results of the model of choice. In particular, we generate full b-ary trees with branching factor $b \in \{2, 4, 6\}$ and

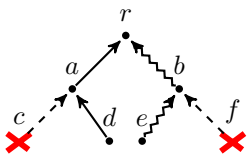


Figure 4: An example of a hierarchical tree with $b = 2$ and $h = 2$ where the generalization paths for d (straight edges) and e (swiggly edges) are represented.

height of $h \in \{2, 3, 4, 5\}$. For each generated tree and for each of its leaf nodes, we generate their generalization paths which will all have the same length k. Then, we then expect that all the nodes in the generalization path should be all ranked in the first k positions with respect to the leaf node. With reference to Figure 4, the leaves are c, d, e, and f: given $h = 2$, all the generalization paths from the leaves will have a length of $k = 2$, and we expect that the distance of a leaf node from the root r is always greater than the distance from any of its siblings or any node that is not within the hierarchy. Given $h = 2$, then all the paths will have length $k = 2$ and will contain 3 candidates: d, a and r for d and e, b and r for e. c and f (crossed out) shall appear as candidates for neither d nor e. Then, we compute the average for all the aforementioned metric evaluated over each generalization path.

Experiment Setup. For *RV/LD/MD**[9], we used all three configurations described by the authors, namely Relevance Vectors, Local Density Vectors with $\beta = 0.75$ and two different configurations for Multiple Descent Vectors: one suggested in the authors' paper ($\beta = 1, \alpha = 0.5$) and the other one is a compromise with the former Local Density Vectors configuration ($\beta = 0.75, \alpha = 0.5$).

For the Poincarré Embeddings, we generate the dataset to be trained using the authors' implementation, and then we retrieve the resulting embeddings and use them in our test sets. We return embeddings with 7 (**P7**) and 50 (**P50**) dimensions and trained over 300 epochs, 50 negative samples and batch size of 10.

For *E*, we train the model with 7 (**E7**) and 50 (**E50**) dimensions. In our repository, we adapted the source code provided by the authors to our use case scenario, so that (1) the category DAG corresponds to the Hierarchy DAG, (2) each category contains only one entity, and (3) the negative examples, in the absence of infrequent words within the same category, are determined by all those elements that are not among the progenitors or the node's successors. Given that the authors already implemented a top-k search in their codebase, we exploit this implementation for returning all the elements that are expected to share a same generalization path.

Finally, since Par2Hier approach necessarily requires a starting vector representation for the terms in the hierarchy, we used as input the vector representation proposed by this article (EE), and then replaced the hierarchical organization of the text in sentences, paragraphs, and documents with the automatically generated taxonomy.

Please note that our embeddings's dimensionality will vary with the branching factor, thus obtaining just one vector per node with dimensionality at most $b = 6$. For all the competitors, we used their metric of preference d alongside their embedding $\mathbf{v}$ (see §2), and benchmarked the solutions over the same accuracy measures. Please observe that our competitors will always have an embedding size greater than ours.

Results. The results are provided in Table 3, 4 and 5. The two most flawed competitors are E (**E7** and **E50**) and both implementations of Par2Hier (**PHC**): in fact, both always provide non zero values of Recall@$> k$, thus showing that such techniques can reconstruct neither smaller nor bigger datasets. In addition to that, E provides a zero Spearman correlation index for small datasets and near zero values for bigger datasets, thus making it inadequate to infer the node distance from their embeddings. Also, E50 takes more than one day to train the dataset with $b = 6$ and $h = 5$, thus making it an unsuitable configuration for training multidimensional representations; those tests were therefore skipped (dashed values in the tables). Albeit Poincarré Embeddings (**P7** and **P50**) provide remarkably good precision and recall values even for bigger datasets, the proposed metric cannot discriminate the nodes that are within a generalization path from the rest, thus making it a non suitable candidate for exact reachability queries on hierarchical data. In fact, for bigger datasets such embedding provides non-zero Recall@$> k$ and nearly zero values for the Spearman correlation index. Last, our benchmaks validate the theoretical accuracy of our model, that is always providing the expected values for Precision@$\leq k$, Recall@$> k$ and Spearman correlation index.

We also provided a time benchmark where we compared our proposed approach to the most accurate competitor, i.e., Poincarré Embeddings. The results of the other competitors are omitted from Table 6, as we prefer to compare the efficiency for the most accurate embedding proposal: while both metrics show comparable running times for smaller datasets, our metric is more efficient by at most one order of magnitude than the Poincarré Embeddings' for $b = 6$ and $h = 5$. In fact, while Poincarré Embeddings require to compute two vectorial norms and an acosh, our Euclidean distance based metric requires to compute just one vectorial norm.

5.2 WordNet's Mammal Hierarchical DAG

Dataset. Albeit we attempted to do some tests over the whole WordNet dataset as in the Poincarré Embeddings, the non-learning competitors' vectorial representation of the whole dataset didn't fit in 64Gb of RAM, and learning-based approaches didn't terminate the preliminary data loading phase after 1 hour. Therefore, we focused on the mammals subset that is generated by a Python script over WordNet in the Poincarré Embeddings' code base. The resulting Hierarchical DAG has a maximum height of 10, and each node has at most two ancestors. It contains 1170 nodes, where the 99% of them had one ancestor. The maximum branching factor of the tree is $b = 35$, thus our EE uses 35 dimensions for representing the embeddings and 99% of the nodes will be represented by just one single vector. For each each leaf of the Mammal Hierarchical DAG loaded using the Lemon Graph Library [5], we generate all the shortest paths towards the root using the same library: this approach guarantees that the positive examples P are generated by a third party tool that is not one of the competitors. Given k the length of each shortest path, we expect that all the nodes in the generalization path of each leaf node should be all ranked in the

[9] MD* stands for MD1 and MD2

Table 3: Average Precision@≤ k over the full tree dataset. Bold (red) represents the best (and worse) values.

Branching	Height	EE	P7	P50	RV	LD	MD1	MD2	PHC	PHS	E7	E50
2	2	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$8.75 \cdot 10^{-1}$	$6.25 \cdot 10^{-1}$	$7.50 \cdot 10^{-1}$	$8.75 \cdot 10^{-1}$	$4.38 \cdot 10^{-1}$	$5.83 \cdot 10^{-1}$	$6.67 \cdot 10^{-1}$	$6.67 \cdot 10^{-1}$
2	3	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$7.50 \cdot 10^{-1}$	$7.75 \cdot 10^{-1}$	$7.75 \cdot 10^{-1}$	$7.75 \cdot 10^{-1}$	$1.93 \cdot 10^{-1}$	$4.65 \cdot 10^{-1}$	$5.94 \cdot 10^{-1}$	$7.50 \cdot 10^{-1}$
2	4	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$6.67 \cdot 10^{-1}$	$6.67 \cdot 10^{-1}$	$7.33 \cdot 10^{-1}$	$7.33 \cdot 10^{-1}$	$1.35 \cdot 10^{-1}$	$3.60 \cdot 10^{-1}$	$3.88 \cdot 10^{-1}$	$7.00 \cdot 10^{-1}$
2	5	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$6.19 \cdot 10^{-1}$	$6.90 \cdot 10^{-1}$	$5.83 \cdot 10^{-1}$	$6.67 \cdot 10^{-1}$	$1.03 \cdot 10^{-1}$	$2.47 \cdot 10^{-1}$	$4.22 \cdot 10^{-1}$	$5.36 \cdot 10^{-1}$
4	2	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$7.92 \cdot 10^{-1}$	$6.46 \cdot 10^{-1}$	$8.54 \cdot 10^{-1}$	$6.46 \cdot 10^{-1}$	$2.29 \cdot 10^{-1}$	$3.50 \cdot 10^{-1}$	$3.13 \cdot 10^{-1}$	$6.67 \cdot 10^{-1}$
4	3	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$5.45 \cdot 10^{-1}$	$5.45 \cdot 10^{-1}$	$5.54 \cdot 10^{-1}$	$5.00 \cdot 10^{-1}$	$7.87 \cdot 10^{-2}$	$2.93 \cdot 10^{-1}$	$3.75 \cdot 10^{-1}$	$5.00 \cdot 10^{-1}$
4	4	$\mathbf{1.00 \cdot 10^{0}}$	$9.98 \cdot 10^{-1}$	$\mathbf{1.00 \cdot 10^{0}}$	$5.18 \cdot 10^{-1}$	$4.75 \cdot 10^{-1}$	$4.43 \cdot 10^{-1}$	$4.43 \cdot 10^{-1}$	$5.99 \cdot 10^{-2}$	$1.60 \cdot 10^{-1}$	$2.59 \cdot 10^{-1}$	$4.09 \cdot 10^{-1}$
4	5	$\mathbf{1.00 \cdot 10^{0}}$	$9.73 \cdot 10^{-1}$	$9.77 \cdot 10^{-1}$	$3.73 \cdot 10^{-1}$	$4.07 \cdot 10^{-1}$	$1.82 \cdot 10^{-1}$	$2.91 \cdot 10^{-1}$	$4.59 \cdot 10^{-2}$	$6.12 \cdot 10^{-2}$	$2.34 \cdot 10^{-1}$	$3.58 \cdot 10^{-1}$
6	2	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$\mathbf{1.00 \cdot 10^{0}}$	$8.33 \cdot 10^{-1}$	$6.24 \cdot 10^{-1}$	$7.21 \cdot 10^{-1}$	$6.10 \cdot 10^{-1}$	$1.81 \cdot 10^{-1}$	$2.56 \cdot 10^{-1}$	$4.26 \cdot 10^{-1}$	$5.28 \cdot 10^{-1}$
6	3	$\mathbf{1.00 \cdot 10^{0}}$	$9.95 \cdot 10^{-1}$	$\mathbf{1.00 \cdot 10^{0}}$	$6.30 \cdot 10^{-1}$	$5.49 \cdot 10^{-1}$	$3.84 \cdot 10^{-1}$	$4.24 \cdot 10^{-1}$	$4.79 \cdot 10^{-2}$	$1.97 \cdot 10^{-1}$	$3.03 \cdot 10^{-1}$	$4.85 \cdot 10^{-1}$
6	4	$\mathbf{1.00 \cdot 10^{0}}$	$9.89 \cdot 10^{-1}$	$9.94 \cdot 10^{-1}$	$3.59 \cdot 10^{-1}$	$4.04 \cdot 10^{-1}$	$3.65 \cdot 10^{-1}$	$3.39 \cdot 10^{-1}$	$2.72 \cdot 10^{-2}$	$8.83 \cdot 10^{-2}$	$2.22 \cdot 10^{-1}$	$3.90 \cdot 10^{-1}$
6	5	$\mathbf{1.00 \cdot 10^{0}}$	$9.21 \cdot 10^{-1}$	$9.46 \cdot 10^{-1}$	$3.22 \cdot 10^{-1}$	$3.00 \cdot 10^{-1}$	$1.56 \cdot 10^{-1}$	$1.56 \cdot 10^{-1}$	$2.24 \cdot 10^{-2}$	$2.82 \cdot 10^{-2}$	$1.52 \cdot 10^{-1}$	––

Table 4: Average Recall@> k over the full tree dataset. Bold (red) represents the best (and worse) values.

Branching	Height	EE	P7	P50	RV	LD	MD1	MD2	PHC	PHS	E7	E50
2	2	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$4.38 \cdot 10^{-1}$	$2.92 \cdot 10^{-1}$	$2.50 \cdot 10^{-1}$	$2.50 \cdot 10^{-1}$
2	3	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$9.09 \cdot 10^{-2}$	$4.55 \cdot 10^{-2}$	$4.55 \cdot 10^{-2}$	$4.55 \cdot 10^{-2}$	$3.15 \cdot 10^{-1}$	$1.77 \cdot 10^{-1}$	$1.48 \cdot 10^{-1}$	$9.09 \cdot 10^{-2}$
2	4	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$4.00 \cdot 10^{-2}$	$4.00 \cdot 10^{-2}$	$3.92 \cdot 10^{-2}$	$3.92 \cdot 10^{-2}$	$1.73 \cdot 10^{-1}$	$1.10 \cdot 10^{-1}$	$1.18 \cdot 10^{-1}$	$5.77 \cdot 10^{-2}$
2	5	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$3.54 \cdot 10^{-2}$	$2.65 \cdot 10^{-2}$	$3.57 \cdot 10^{-2}$	$3.51 \cdot 10^{-2}$	$9.45 \cdot 10^{-2}$	$7.46 \cdot 10^{-2}$	$6.09 \cdot 10^{-2}$	$4.88 \cdot 10^{-2}$
4	2	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$1.39 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.39 \cdot 10^{-2}$	$1.10 \cdot 10^{-1}$	$6.56 \cdot 10^{-2}$	$1.15 \cdot 10^{-1}$	$5.56 \cdot 10^{-2}$
4	3	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$9.50 \cdot 10^{-3}$	$9.50 \cdot 10^{-3}$	$3.16 \cdot 10^{-3}$	$9.49 \cdot 10^{-3}$	$4.16 \cdot 10^{-2}$	$2.28 \cdot 10^{-2}$	$3.09 \cdot 10^{-2}$	$2.47 \cdot 10^{-2}$
4	4	$\mathbf{0.00 \cdot 10^{0}}$	$2.33 \cdot 10^{-5}$	$\mathbf{0.00 \cdot 10^{0}}$	$3.74 \cdot 10^{-3}$	$3.75 \cdot 10^{-3}$	$3.01 \cdot 10^{-3}$	$3.01 \cdot 10^{-3}$	$1.23 \cdot 10^{-2}$	$9.88 \cdot 10^{-3}$	$1.10 \cdot 10^{-2}$	$8.79 \cdot 10^{-3}$
4	5	$\mathbf{0.00 \cdot 10^{0}}$	$1.20 \cdot 10^{-4}$	$1.03 \cdot 10^{-4}$	$1.48 \cdot 10^{-3}$	$1.48 \cdot 10^{-3}$	$1.49 \cdot 10^{-3}$	$1.48 \cdot 10^{-3}$	$3.72 \cdot 10^{-3}$	$3.60 \cdot 10^{-3}$	$3.38 \cdot 10^{-3}$	$2.84 \cdot 10^{-3}$
6	2	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$\mathbf{0.00 \cdot 10^{0}}$	$1.67 \cdot 10^{-2}$	$\mathbf{0.00 \cdot 10^{0}}$	$8.33 \cdot 10^{-3}$	$3.82 \cdot 10^{-2}$	$2.84 \cdot 10^{-2}$	$4.31 \cdot 10^{-2}$	$3.54 \cdot 10^{-2}$
6	3	$\mathbf{0.00 \cdot 10^{0}}$	$7.26 \cdot 10^{-5}$	$\mathbf{0.00 \cdot 10^{0}}$	$3.28 \cdot 10^{-3}$	$3.96 \cdot 10^{-3}$	$2.66 \cdot 10^{-3}$	$3.98 \cdot 10^{-3}$	$1.26 \cdot 10^{-2}$	$7.35 \cdot 10^{-3}$	$1.09 \cdot 10^{-2}$	$8.08 \cdot 10^{-3}$
6	4	$\mathbf{0.00 \cdot 10^{0}}$	$3.53 \cdot 10^{-5}$	$1.94 \cdot 10^{-5}$	$8.63 \cdot 10^{-4}$	$7.55 \cdot 10^{-4}$	$6.48 \cdot 10^{-4}$	$6.48 \cdot 10^{-4}$	$2.63 \cdot 10^{-3}$	$2.19 \cdot 10^{-3}$	$2.51 \cdot 10^{-3}$	$1.97 \cdot 10^{-3}$
6	5	$\mathbf{0.00 \cdot 10^{0}}$	$5.96 \cdot 10^{-5}$	$3.45 \cdot 10^{-5}$	$1.97 \cdot 10^{-4}$	$1.97 \cdot 10^{-4}$	$2.15 \cdot 10^{-4}$	$2.15 \cdot 10^{-4}$	$5.38 \cdot 10^{-4}$	$5.29 \cdot 10^{-4}$	$5.46 \cdot 10^{-4}$	––

Table 5: Average Spearman Correlation Index over the full tree dataset. Bold (red) represents the best (and worse) values.

Branching	Height	EE	P7	P50	RV	LD	MD1	MD2	PHC	PHS	E7	E50
2	2	$\mathbf{1.00 \cdot 10^{0}}$	$4.64 \cdot 10^{-1}$	$4.64 \cdot 10^{-1}$	$6.59 \cdot 10^{-1}$	$7.04 \cdot 10^{-1}$	$6.40 \cdot 10^{-1}$	$6.10 \cdot 10^{-1}$	$3.64 \cdot 10^{-1}$	$4.93 \cdot 10^{-1}$	$0.00 \cdot 10^{0}$	$0.00 \cdot 10^{0}$
2	3	$\mathbf{1.00 \cdot 10^{0}}$	$2.01 \cdot 10^{-1}$	$2.01 \cdot 10^{-1}$	$3.71 \cdot 10^{-1}$	$4.02 \cdot 10^{-1}$	$3.68 \cdot 10^{-1}$	$4.17 \cdot 10^{-1}$	$7.40 \cdot 10^{-2}$	$1.60 \cdot 10^{-1}$	$9.38 \cdot 10^{-3}$	$2.14 \cdot 10^{-2}$
2	4	$\mathbf{1.00 \cdot 10^{0}}$	$7.89 \cdot 10^{-2}$	$7.89 \cdot 10^{-2}$	$2.54 \cdot 10^{-1}$	$2.55 \cdot 10^{-1}$	$2.29 \cdot 10^{-1}$	$2.65 \cdot 10^{-1}$	$1.92 \cdot 10^{-2}$	$6.40 \cdot 10^{-2}$	$3.07 \cdot 10^{-3}$	$1.94 \cdot 10^{-2}$
2	5	$\mathbf{1.00 \cdot 10^{0}}$	$2.87 \cdot 10^{-2}$	$2.87 \cdot 10^{-2}$	$1.65 \cdot 10^{-1}$	$1.70 \cdot 10^{-1}$	$1.42 \cdot 10^{-1}$	$1.63 \cdot 10^{-1}$	$5.77 \cdot 10^{-3}$	$2.10 \cdot 10^{-2}$	$2.56 \cdot 10^{-3}$	$6.73 \cdot 10^{-3}$
4	2	$\mathbf{1.00 \cdot 10^{0}}$	$7.15 \cdot 10^{-2}$	$7.15 \cdot 10^{-2}$	$3.71 \cdot 10^{-1}$	$3.81 \cdot 10^{-1}$	$3.41 \cdot 10^{-1}$	$3.86 \cdot 10^{-1}$	$9.64 \cdot 10^{-2}$	$1.51 \cdot 10^{-1}$	$0.00 \cdot 10^{0}$	$0.00 \cdot 10^{0}$
4	3	$\mathbf{1.00 \cdot 10^{0}}$	$7.94 \cdot 10^{-3}$	$7.94 \cdot 10^{-3}$	$1.71 \cdot 10^{-1}$	$1.72 \cdot 10^{-1}$	$1.56 \cdot 10^{-1}$	$1.57 \cdot 10^{-1}$	$8.23 \cdot 10^{-3}$	$1.53 \cdot 10^{-2}$	$3.25 \cdot 10^{-4}$	$7.76 \cdot 10^{-4}$
4	4	$\mathbf{1.00 \cdot 10^{0}}$	$7.63 \cdot 10^{-4}$	$7.63 \cdot 10^{-4}$	$6.57 \cdot 10^{-2}$	$6.74 \cdot 10^{-2}$	$5.73 \cdot 10^{-2}$	$5.81 \cdot 10^{-2}$	$6.75 \cdot 10^{-4}$	$1.45 \cdot 10^{-3}$	$1.80 \cdot 10^{-5}$	$6.85 \cdot 10^{-5}$
4	5	$\mathbf{1.00 \cdot 10^{0}}$	$6.73 \cdot 10^{-5}$	$6.73 \cdot 10^{-5}$	$2.48 \cdot 10^{-2}$	$2.61 \cdot 10^{-2}$	$1.95 \cdot 10^{-2}$	$1.96 \cdot 10^{-2}$	$5.83 \cdot 10^{-5}$	$1.27 \cdot 10^{-4}$	$1.72 \cdot 10^{-6}$	$5.88 \cdot 10^{-6}$
6	2	$\mathbf{1.00 \cdot 10^{0}}$	$1.83 \cdot 10^{-2}$	$1.83 \cdot 10^{-2}$	$2.53 \cdot 10^{-1}$	$2.45 \cdot 10^{-1}$	$2.40 \cdot 10^{-1}$	$2.59 \cdot 10^{-1}$	$7.34 \cdot 10^{-2}$	$8.66 \cdot 10^{-2}$	$0.00 \cdot 10^{0}$	$0.00 \cdot 10^{0}$
6	3	$\mathbf{1.00 \cdot 10^{0}}$	$8.82 \cdot 10^{-4}$	$8.82 \cdot 10^{-4}$	$7.94 \cdot 10^{-2}$	$8.34 \cdot 10^{-2}$	$6.77 \cdot 10^{-2}$	$7.62 \cdot 10^{-2}$	$7.48 \cdot 10^{-3}$	$4.85 \cdot 10^{-3}$	$3.59 \cdot 10^{-5}$	$6.96 \cdot 10^{-5}$
6	4	$\mathbf{1.00 \cdot 10^{0}}$	$3.71 \cdot 10^{-5}$	$3.71 \cdot 10^{-5}$	$2.09 \cdot 10^{-2}$	$2.21 \cdot 10^{-2}$	$1.60 \cdot 10^{-2}$	$1.78 \cdot 10^{-2}$	$2.39 \cdot 10^{-4}$	$2.30 \cdot 10^{-4}$	$9.72 \cdot 10^{-7}$	$2.72 \cdot 10^{-6}$
6	5	$\mathbf{1.00 \cdot 10^{0}}$	$1.45 \cdot 10^{-6}$	$1.45 \cdot 10^{-6}$	$5.01 \cdot 10^{-3}$	$5.55 \cdot 10^{-3}$	$3.46 \cdot 10^{-3}$	$3.88 \cdot 10^{-3}$	$6.62 \cdot 10^{-6}$	$9.49 \cdot 10^{-6}$	$1.67 \cdot 10^{-8}$	––

Table 6: Average speed in ms between the two best competitors over the full tree dataset.

Branching	Height	EE	P50	P7
2	2	$3.59 \cdot 10^{-2}$	$\mathbf{3.33 \cdot 10^{-2}}$	$3.77 \cdot 10^{-2}$
2	3	$5.91 \cdot 10^{-2}$	$\mathbf{4.97 \cdot 10^{-2}}$	$\mathbf{4.97 \cdot 10^{-2}}$
2	4	$\mathbf{4.80 \cdot 10^{-2}}$	$8.80 \cdot 10^{-2}$	$8.29 \cdot 10^{-2}$
2	5	$\mathbf{8.89 \cdot 10^{-2}}$	$1.45 \cdot 10^{-1}$	$1.80 \cdot 10^{-1}$
4	2	$\mathbf{8.08 \cdot 10^{-2}}$	$1.12 \cdot 10^{-1}$	$1.27 \cdot 10^{-1}$
4	3	$\mathbf{2.94 \cdot 10^{-1}}$	$3.51 \cdot 10^{-1}$	$3.22 \cdot 10^{-1}$
4	4	$1.67 \cdot 10^{0}$	$\mathbf{1.37 \cdot 10^{0}}$	$1.43 \cdot 10^{0}$
4	5	$\mathbf{4.43 \cdot 10^{0}}$	$5.60 \cdot 10^{0}$	$6.31 \cdot 10^{0}$
6	2	$\mathbf{2.63 \cdot 10^{-1}}$	$2.94 \cdot 10^{-1}$	$3.01 \cdot 10^{-1}$
6	3	$\mathbf{1.23 \cdot 10^{0}}$	$1.40 \cdot 10^{0}$	$1.59 \cdot 10^{0}$
6	4	$\mathbf{8.17 \cdot 10^{0}}$	$1.25 \cdot 10^{1}$	$2.27 \cdot 10^{1}$
6	5	$\mathbf{3.95 \cdot 10^{1}}$	$1.63 \cdot 10^{2}$	$1.48 \cdot 10^{2}$

first $k+1$ positions. Then, similarly to the previous experiment, we compute the average for all the aforementioned accuracy metrics outcomes evaluated over each generalization path.

Experimental Setup. For Direct Acyclic Graphs, we discard the two worst competitors that showed major flaws in the stress tests, and focus on Poincarré Embeddings and RV/LD/MD*, where the former requires a preliminary training phase, while the latter does not. For these competitors, we used the same experiment settings as for the full tree dataset: given that RV/LD/MD* are only defined over Hierarchical Tree, we use Algorithm 1 to generate multiple possible embeddings for each node and then, given Definition 4.3 and that similarity is the dual definition of a distance, we use as a similarity metric the following:

$$\mathrm{sim}(n, m) = \max_{n' \in \mathcal{G}(n), m' \in \mathcal{G}(m)} \cos(n', m')$$

Table 7: Qualitative and Performance benchmark over WordNet's Mammals. Bold (red) represents the best (and worse) values.

Metrics	EE	P50	RV/LD/MD*
Avg. Precision@≤ k	**1**	$9.35 \cdot 10^{-1}$	$5.80 \cdot 10^{-3}$
Avg. Recall@> k	**0**	$3.38 \cdot 10^{-4}$	**0**
Avg. Spearman Correlation	**1**	$1.20 \cdot 10^{-4}$	0
Time (ms)	$1.45 \cdot 10^{-4}$	$1.52 \cdot 10^{-4}$	$\mathbf{1.09} \cdot 10^{-4}$

where cos is the authors' cosine similarity that we also adopted in the previous experiment, and the dual problem of distance minimization is similarity maximisation.

Results. Table 7 provides the outcome for this last experiment. All the density-based vector embeddings proposed by [13] (i.e., RV/LD/MD*) provide the same results: albeit we adopted for this representation a similar strategy as the one from our proposal (d^G), all these embeddings fail in providing a correlation with the node ranking expected from the minimum path from the leaf node towards the root. This consideration in combination with the Recall@> k being zero implies that the metric fails to reconstruct the exact path's node sequence even though there are no false negatives. On the other hand, Poincarré with 50 dimensions fails to meet this last requirement: this consideration jointly with the nearly optimal results of Precision@≤ k confirms the observations done in the Introduction, that is that some sibling nodes might be actually nearer to each other than the distance between them and their first ancestor. Given that the Poincarré distance has no threshold value, it cannot separate the nodes within the same hierarchy path from the others, thus providing a low (but non-zero) score on the Spearman correlation.

Last, our metric joined with our embedding proposal outperforms in precision the other two competitors, thus empirically showing the correctness of our implemented solution. Even in this case, our proposed metric proves to be more efficient than the most precise competitor.

6 CONCLUSION AND FUTURE WORKS

This paper presents a novel approach for embedding hierarchical data without the need of a preliminary training phase. The problem is approximated (i.e., multiple vectors might be returned) instead of approximating the solution (i.e., distance preservation). After providing some intuitive proof of correctness of our proposed approach, we give some benchmarks with state of the art hierarchical embeddings: from these results, we show that our technique outperforms the state-of-the-art approaches both in accuracy and in running time. On the other hand, the competing approaches fail to always provide exact solutions over different data distributions and sizes, which is crucial for indexing multidimensional data.

We plan to extend the present work in three ways. First, given that hierarchical data resulting from data integration and linking processes comes with uncertainty values [20], we should extend the current embeddings so to also consider those values as part of the metric. Second, we should generalize the proposed approach to direct graphs that might contain loops. Real world semantic graphs such as ConceptNet [20] and WordNet [14] might contain synonymy and relatedness relationships creating loops. These relationships are also worth considering when semantic datasets are the result of imprecise data integration processes, where not all the concepts are directly associated to a generalization path. Albeit it is trivial to represent one directed graph with cycles with multiple Hierarchical DAGs which can be then converted into multiple Hierarchical Trees, this approach will not scale up as we will potentially obtain an exponential number of vectors associated to each node of a graph. In our future work, we will investigate how graph summarizations can be beneficial to group related concepts together when synsets are not available [20] and if it is always possible to represent complex semantic datasets as DAGs with some degree of approximation. Last, we will need also to investigate which is the accuracy loss provided by dimensionality reduction techniques for vectorial data representations such as Multi-Dimensional Scaling [22]. Our future work will also extend the formalization section, thus providing formal proofs of correctness for our definitions.

REFERENCES

[1] G. Bergami. A framework supporting imprecise queries and data. *CoRR*, abs/1912.12531, 2019.
[2] F. Bertini, G. Bergami, D. Montesi, G. Veronese, G. Marchesini, and P. Pandolfi. Predicting frailty condition in elderly using multidimensional socioclinical databases. *Proc. of the IEEE*, 106(4):723–737, April 2018.
[3] P. Ciaccia, M. Patella, and P. Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *PVLDB*, page 426–435, 1997.
[4] L. De Raedt. *Logical and Relational Learning*. Springer Inc., 1st edition, 2010.
[5] B. Dezso, A. Jüttner, and P. Kovács. LEMON - an open source C++ graph template library. *Electron. Notes Theor. Comput. Sci.*, 264(5):23–45, 2011.
[6] J. Grant and M. V. Martinez. *Measuring Inconsistency in Information*. College Publications, 2018.
[7] G. Guennebaud, B. Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.
[8] Haixun Wang, Hao He, Jun Yang, P. S. Yu, and J. X. Yu. Dual labeling: Answering graph reachability queries in constant time. In *ICDE*, pages 75–75, April 2006.
[9] A. Hotho. Data mining on folksonomies. In *Intelligent Information Access*, pages 57–82. 2010.
[10] Z. Hu, P. Huang, Y. Deng, Y. Gao, and E. P. Xing. Entity hierarchy embedding. In *ACL*, pages 1292–1300, 2015.
[11] J. Johnson. *Hypernetwors in the Science of Complex Systems*. Imperial College Press, London, UK, UK, 2011.
[12] A. Klimovskaia, D. Lopez-Paz, L. Bottou, and M. Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature Communications*, 11(2966), 2020.
[13] H. Liu, H. Bao, and D. Xu. Concept vector for similarity measurement based on hierarchical domain structure. *Computing and Informatics*, 30(5):881–900, 2011.
[14] G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.
[15] M. Nickel and D. Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6338–6347, 2017.
[16] A. Petermann, G. Micale, G. Bergami, A. Pulvirenti, and E. Rahm. Mining and ranking of generalized multi-dimensional frequent subgraphs. In *ICDIM*, pages 236–245, Sep. 2017.
[17] J. B. Saxe. Embeddability of weighted graphs ink-space is strongly np-hard. In *Proceedingsof 17th Allerton Conference in Communications, Control and Computing*, 1979.
[18] C. Seshadhri, A. Sharma, A. Stolman, and A. Goel. The impossibility of low-rank representations for triangle-rich complex networks. *Proc. of the National Academy of Sciences*, 117(11):5631–5637, 2020.
[19] K. Simon. An improved algorithm for transitive closure on acyclic digraphs. *Theor. Comput. Sci.*, 58:325–346, 1988.
[20] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proc. of AAAI*, pages 4444–4451, 2017.
[21] T. Teofili. par2hier: towards vector representations for hierarchical content. In *ICCS 2017, 12-14 June 2017, Zurich, Switzerland*, pages 2343–2347, 2017.
[22] J. Wills, S. Agarwal, D. Kriegman, and S. Belongie. Toward a perceptual space for gloss. *ACM Trans. Graph.*, 28(4), Sept. 2009.
[23] J. X. Yu and J. Cheng. Graph reachability queries: A survey. In *Managing and Mining Graph Data*, pages 181–215. 2010.

Guidelines for Cybersecurity Visualization Design

Younho Seong*, Joseph Nuamah**, & Sun Yi***
*Industrial & Systems Engineering Dept. NC A&T SU
**School of Industrial Engineering & Management, Oklahoma State University
***Mechanical Engineering Dept. NC A&T SU
Greensboro, NC, USA
yseong@ncat.edu

ABSTRACT

Cyber security visualization designers can benefit from human factors engineering concepts and principles to resolve key human factors challenges in visual interface design. We survey human factors concepts and principles that have been applied in the past decade of human factors research. We highlight these concepts and relate them to cybersecurity visualization design. We provide guidelines to help cybersecurity visualization designers address some human factors challenges in the context of interface design. We use ecological interface design approach to present human factors-based principles of interface design for visualization. Cyber security visualization designers will benefit from human factors engineering concepts and principles to resolve key human factors challenges in visual interface design.

CCS CONCEPTS

• **Human-centered computing**;

KEYWORDS

Cybersecurity, Visualization, Ecological Interface Design, Affordance, Cognition

ACM Reference Format:
Younho Seong*, Joseph Nuamah**, & Sun Yi***. 2020. Guidelines for Cybersecurity Visualization Design. In *Proceedings of 24th International Database Engineering & Application Symposium (IDEAS 2020).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410566.3410596

1 INTRODUCTION

Cybersecurity, which includes information assurance [1], refers to technologies, processes and practices used to protect information and systems from unauthorized access. According to [2] (p.46), "cybersecurity is essentially a human-on-human adversarial game played out by automated avatars". On the one hand, human cyber attackers with malicious intent relentlessly attempt to outsmart defensive measures, attack systems and wreak havoc. On the other hand, human cyber defenders collaborate with computers to monitor and thwart these cyberattacks. In particular when the situation becomes very complex, human cyber defenders and computers work together to discover, resolve, and respond to malicious activities [3]. Computers are designed to process massive volumes of data within a relatively short time. However compared to humans, computers find it difficult detecting complex patterns. Traditional visualization tools are designed in such a way that they either accentuate the strengths of computers or their human operators, but not both. Humans are highly effective and fast at making sense of partial, incomplete or rapidly presented information, but are unable to cognitively process large amounts of rapidly changing information. Computers, on the other hand, have rapid data processing capabilities, but are not effective at drawing inferences from incomplete information. Consequently, it is needful to combine the fast data processing capabilities of computers with intuitive decision making skills of humans [4] to improve human information throughput and decision making. According to [5], they defined information visualization as the art and science of representing abstract information in a visual form that enables human users to gain insight through their perceptual and cognitive capabilities. Cyber defense depends on visual interfaces as the single point of connection to cyberspace. Thus, visualization allows cyber defenders to interact with communication networks, network devices, and cyber-physical systems. However, visualization will be of little value if it is too complex, difficult to navigate, does not tie information to present or future goals, or if cyber defenders are overloaded and exhausted [6]. The human factor requires more attention in cyber security visualization ([7]; [8]). Cyber security visualization designers can benefit from proven human factors concepts and principles to resolve key challenges in visual interface design. In this paper, we survey human factors concepts and principles that have been applied in the past decade of human factors research. We seek to highlight these concepts and relate them to cyber security visualizations. Finally, we provide guidelines to help cybersecurity visualization designers address some human factors issues in their designs.

2 RELATED WORKS

Previous researchers have identified challenges with cybersecurity visualizations. For example, [9] identified seven challenges that must be considered when developing cyber security visualization. These are (i) volume, variety, and velocity of data (big data), (ii) multiple data sources, (iii) unlinked data sources, (iv) quality of data, (v) pattern of network, (vi) threat escalation progression, and (vii) balancing risk and reward. They argued that visualization should first present cyber security defenders with a majority of the important

IDEAS 2020, August 12-14, 2020, Seoul, South Korea

ACM ISBN 978-1-4503-7503-0/20/06.
https://doi.org/10.1145/3410566.3410596

information they need using a minimalist approach. Subsequent information presentation should support the ability to explore further if they choose to do so. Visualization should be able to integrate multiple data sources, and tie information together. Unlinked data sources could be linked visually through parameters such as time, IP address or port number. Data quality issues including missing or corrupt data and confidence in correctness of data should be accounted for in the visualization. Analysts should be able to understand the network status through network cadence presented by the visualization. The visualization should be able to present to analysts information about potential risks or rewards for a current line of inquiry [9]. In [6], they mentioned cyber-cognitive situation awareness, a phrase coined by [10], and interface design as areas likely to benefit cyber defense operations. Cyber defenders endeavor to achieve and maintain situational awareness of the networks they defend through visualizations and visual analytics. Additionally, using level of situation awareness [11], cyber defenders' must invest significant cognitive resources in anticipatory cognition in order to draw inferences about what will happen next. They perceive critical network information, understand the perceived information, and then take action or predict the environment [11]. In [6] they argued that following human factors design principles should make visualizations more valuable to cyber security defenders. Until recently, the human element had been neglected in cyber defense. In [12], they emphasized the need for cyber defenders to be able to perceive and comprehend disparate network information elements so they can determine the status of the networks they monitor. Designers of cyber security visualization need not simply present all of the possible information from the system. Rather, they should include the needs of human operators and have a system to account for the dynamics of both [12].

2.1 Information Visualization and Cognition

The tremendous amount of data generated by sensor networks makes cyber defenders susceptible to information overload ([9]; [6]). An appreciation of how humans perceive and process visual information will enable us to understand how, for example, shapes and colors communicate large volumes of data more effectively than tables of numbers or paragraphs of text. Researchers have employed dual process theories to explain how humans perceive and process visually. The theories supported by much evidence in cognitive science [13] have been the focus of contemporary research ([14]; [15]; [16]) and have had various labels attached to each one. Theorists assume that cognitive tasks evoke two types of decision making processes: intuition (System 1) and analysis (System 2). Intuition comprises subconscious processing. Analytical cognition is conscious, effortful, and requires cognitive resources. Intuitive cognition is automatic, nonverbal, associative, rapid, effortless, concrete, holistic, and requires minimal cognitive resources. Information may be presented in a format that elicits intuition or analytical cognition. Information visualization should seek to encode information in such a way that it will be automatically and correctly perceived by system 1, thus freeing up the system 2 for more involved understanding of data.

2.2 Principles of Display Design

Visualization design must conform to principles that engender efficient and effective perception, comprehension, decision making. In this discussion, we focus on principles rather than guidelines, since principles are supported by empirical research. In [18], they presented 13 display design principles and grouped them into perceptual principles, mental model principles, principles based on attention, and memory principles. These principles may be tailored to specific situations, and may be utilized to create effective visualizations. It behooves the visualization designers to apply and harmonize them in order to guarantee the appropriate level of situation awareness.

Perceptual Principle Objects that are displayed should be discernable for the human operator to effectively use them (make displays legible). Operators should not be made to judge the represented variable level based on a single sensory dimension. Visualization designers should use multiple parameters to code objects (avoid absolute judgment levels). Operators perceive and interpret signals based on their past experience. More physical evidence of a signal must be presented to ensure that it is interpreted correctly, if that signal is as a result of some unlikely event, contrary to the expectations of the human participant (top-down processing). There is a greater chance that a signal presented in alternative physical forms (e.g., color and shape) will result in likelihood of it being perceived and understood (redundancy gain). To prevent the confusion that arises from similar appearing signals, designers should accentuate features that are dissimilar for such signals and delete features that are unnecessary (discriminability).

Mental Model Principles Designers should design visualizations to capture and match the mental models of operators. Displays should look like the variables they represent. Elements should be configured in the same manner as the environment they represent (principle of pictorial realism from Gestalt theory). For dynamic displays, moving elements should be shown to move in a pattern and direction compatible with the operator's mental model (principle of the moving part).

Principles Based on Attention Certain information should require minimal effort to access. The visualization should be designed to minimize the cost in time or effort to access pertinent information (minimize information access cost). In cybersecurity, cyber defenders obtain more than one piece of information from multiple displays. Attention must be given to these pieces of information for the same purpose. Fortunately, humans do not perceive only spatial proximity. We also perceive proximity in color, time, intensity, and shape. Thus, visualization designers should decrease information access costs by using properties of objects to organize related objects to be near each other in perceptual space (proximity compatibility principle). This allows operators to easily access the objects at the same time. Operators find it difficult perceiving and processing lots of information at the same time. This is a human limitation, which can be overcome by dividing information across multiple resources. Therefore, dividing information between modalities say, visual and auditory, is more effective than presenting it only visually or auditorily (principle of multiple resources).

Memory Principles Operators need not retain information solely in working memory or retrieve information solely from long-term

memory during task execution. Information should be made visible when needed to minimize reliance on memory. Effective visualizations balance the use of knowledge in the head of the operator and knowledge in the world (replace memory with visual information). Designers should seek to design visualizations that eliminate resource-demanding cognitive tasks and replace them with simpler perceptual tasks. Predictive displays allow operators to not only focus on current conditions, but also think about possible future conditions. This is because these displays are able to replace resource-demanding cognitive tasks with simple perceptual displays, thereby reducing cognitive workload on human operators. Visualizations that project into the future allow operators to be more proactive than reactive (principle of predictive aiding). System designers can take advantage of the human long term memory to work too well, and automatically making human operators take actions that are not needed. Visualization designers should design visualizations that are consistent with other visualizations operators concurrently use or have used in the past (principle of consistency).

2.3 Pre-attentive Processing

How humans perceive visual information is key during visualization design ([48]; [20]). Pre-attentive processing refers to the way humans effortlessly and automatically categorizes images in a visual field [20]. Humans have the innate ability to recognize visual attributes such as shapes, edges, color, patterns, and motion - referred to as pre-attentive attributes [21]. In [48] it presents four pre-attentive properties: color, form, movement, and spatial positioning that can be used to encode data. Theories that explain pre-attention processing include feature integration, and guided search. The feature extraction theory has two stages: pre-attentive and focused attention. The first stage, pre-attentive stage, is a parallel process in which basic features in the visual field are automatically and effortlessly gathered. This first stage is fast, and does not depend on attention. The second stage, focused attention, is a serial process in which individual features are combined to perceive the whole object. Stored knowledge may influence feature combination [21]. The feature integration theory assumes that the first stage is entirely parallel, and the second stage is serial. It ignores the effect of similarity between stimuli and target. According to the guided search theory [22], pre-attentive information from the early stage is used to guide attention in the later stage. Visual search efficiency is associated with the effectiveness of the guidance, which may range from perfect guidance (feature pop-out) to no guidance (no basic features distinguish target from non-targets). Designers of cybersecurity visualization should seek to visually encode information preattentively. They should encode data using pre-attentive attributes if they want operators to perceive them instantaneously. Furthermore, they should use different pre-attentive attributes if they want some of the data to stand out from the rest.

2.4 Gestalt Principles for Visualization

Humans are able to group separate visual objects into a cognitive unit, and this ability may be exploited to influence visualization design [23]. The gestalt theory explains how human perception groups in an effort to make sense of what we perceive. Grouping is the key process in human visual perception [24]. Gestalt principles may be used as visualization strategies [25]. The principles help us understand the different factors that influence the grouping of elements into a whole. They include proximity, similarity, connectedness, symmetry, closure, common fate, good continuation, past experience, prägnanz, and smoothness. The principles of proximity, similarity, connectedness, continuity, and common fate define how visual elements are grouped together to form coherent wholes [26]. The proximity principle states that objects near to each other are perceived as a group. Humans perceive objects that are further apart as less related. The similarity principle states that objects that are similar are perceived as a group. Designers should use similar attributes to establish relationships between visual objects in order to reinforce groupings. The connectedness principles states that objects connected by other elements are perceived as a group. The continuity principle states that objects that form continuous curves are more likely to be perceived as a group. To facilitate comparison of different objects, for example, designers could arrange visual elements on a line or a curve. The common fate principle states that when objects are aligned or move the same direction they are perceived as a group. Visualization designers can use direction to establish or negate relationships. The closure principle states that a group of elements surrounded by a visual element is perceived as a group. Visualization designers should ensure that there is enough contrast between figures and ground. The figure-ground principle states that visual elements are perceived as either distinct elements of focus or the background on which figures rest. The prägnanz principle states that humans tend to process simple patterns faster than complex patterns. Thus, visualization designers should arrange visual objects logically. Previous researchers have applied the gestalt principles in their work. In [49], for instance, they applied seven gestalt principles (simplicity, familiarity, similarity, continuation, proximity, common fate) to animated visualizations of network data to aid network operators understand structural and visual changes. In [50], they demonstrated the effects of gestalt principles on graph comprehension. In [27], they employed four gestalt principles, proximity, similarity, prägnanz and common fate, to develop a user-centric visualization.

2.5 Affordance

In [28], he first used the term "affordance" to mean attributes of an interface or device that suggest how it is used. Also in [29] he qualified the term "affordance" and suggested two kinds of affordance: real affordance and perceived affordance. He argues that there can be both real and perceived affordances (the two are not necessarily the same) when users deal with real, physical objects. In [30], it expanded the works of Gibson and Norman by dividing affordance into four types: cognitive affordance (same as Norman's perceived affordance), physical affordance (same as Norman's real affordance), sensory affordance, and functional affordance. A cognitive affordance assists knowledge of something or thinking about it. A physical affordance assists physical access to something. A sensory affordance assists the operator to sense something. A functional affordance assists operators do something in the work domain. All four affordances (cognitive, physical, sensory, and functional) should be considered together in any visualization design

[30]. Affordance-based design is accepted as a formal design approach [30];[31]. In [32] they employed the four types of affordance in the design of cybersecurity control room visualization. They employed cognitive affordance by separating network status data with different colors and screen locations. They implemented physical affordance by making the display respond to users' movements towards the screen. In their design, sensory affordance enabled operators to see and discriminate different visual elements. They employed functional affordance by providing several operators the ability to zoom, explore, discuss, and assign tasks [32].

2.6 Cognitive Task Analysis

In [33], (p. 168) they defined as cognitive task analysis (CTA) as "a set of methods to elicit, explain, and represent the mental processes involved in performing a task". Researchers have employed CTA to enable them develop situation awareness aids for cybersecurity operators. For example in [34] they employed CTA to gain insight into the work processes, cognitive skills, and tools that cybersecurity defenders rely on to achieve situation awareness. They advised that visualizations should be aligned to cyber defenders' roles. In [35], they employed CTA to gain insight into situation awareness requirements for a cyber situation awareness tool. They developed scenarios, interviewed a subject matter expert (SME) from whom they established a preliminary list of cognitive tasks. Then, they classified the cognitive tasks into types of awareness that support cyber situation awareness.

2.7 Ecological Interface Design

Two main approaches to interface design are user-centered design (UCD), and ecological interface design (EID). On the one hand, UCD focuses on the capabilities and limitations of human operators, and seeks to amplify and extend their perceptual, cognitive, and performance capabilities ([36]; [37]). On the other hand, EID focuses on the work domain, and seeks to design tools that support human operators by leveraging their perception, action, or cognitive capabilities ([38]; [39]; [40]). The UCD attempts to identify information that is needed to complete only certain well-defined tasks. EID proposes that the system must be understood first, and then displayed in a way that is useful for the user [41]. We agree with [42] (p.295) that EID is a more comprehensive framework within which the UCD can play an important supporting role. EID approach is very fitting for the cyber defense environment. Here, the visual display stands between the cyber defender and work domain. The success of the visualization depends on the compatibility between the work domain and the visual display, and compatibility between the display and the cyber defender ([40]; [17]).

Work domain analysis involves building a functional representation of the system under analysis. EID uses abstraction and aggregation hierarchies as analytical tools for understanding a particular work domain. Abstraction hierarchy models domain constraints in terms of 'means-ends'. It organizes physical resources and system functions into five levels, with each level representing work domain properties and related information. The objective of abstraction hierarchy is to clearly define levels, and operational constraints for each level, so that the human operator knows precisely when and where a constraint breaks if an abnormality occurs. Aggregation hierarchy provides 'part-whole' models of the work domain by decomposing or aggregating items on each level of the abstraction hierarchy. A model of the structure of the work domain is obtained by using abstraction and aggregation hierarchies in work analysis [40].

The thrust of EID is to make processes more transparent and observable to the human operator, so that they can use lower level perceptual processing rather than higher level analytical processing. It uses the skill, rule, and knowledge (SRK) framework to describe decision making processes that human operators adopt based on their level of expertise and the decision situation [44]. The SRK taxonomy supports skill-based and rule-based behavior for familiar or routine tasks. This enables operators to devote more cognitive resources to knowledge-based behavior for detection of anomalies or unexpected events.

In [1] he presented three EID design principles, direct perception, direct manipulation and visual momentum, which can be applied to achieve EID goals. The direct perception principle is applied when the visual display matches both the perceptual skills of the cyber defender and the specific demands of the work. This occurs when the work domain affordances are all together visible in the interface, and cyber defenders are able to perceive these affordances through consistent spatio-temporal patterns in the visual display (Bennett, 2014a). Direct manipulation refers to the extent to which system controls allow the defender to execute input directly via controls in the interface [1](p.1235). This means the cyber defender is able to act directly upon objects of interest on the display. Cyber defenders are often required to navigate through multiple display screens and windows in order to integrate information [2]. They might 'get lost', and therefore not know where they are or find it difficult to decide where to go next 1 in a network of displays. Also, the information required by the cyber defender cannot all be displayed in parallel. In this case, the cyber defender may not be able to locate information of interest. This is known as the keyhole effect. In order to prevent keyhole effects and the getting lost phenomenon, In [45] he proposed the concept of visual momentum. The visual momentum principle refers to "... the extent to which an interface supports a practitioner in transitioning between various perceptual and cognitive information-seeking activities that are required for understanding and exploring work domains?" [43] (p. 399).

In [41] they employed EID to design visualization for monitoring network performance and availability. The tool they developed was more effective and accurate in diagnosis that HP Open View Network Node Manager. In [1] he used EID to develop VEILS (Versatile Ecological Interface for Lockdown network Security), an ecological interface for cyber network defense. VEILS was designed based on three EID principles: direct perception, direct manipulation and visual momentum.

3 GUIDELINES FOR CYBERSECURITY VISUALIZATION

Much of the cybersecurity literature contains evaluation of visualizations [46]; [47]. For example in [47], they discuss gaps in evaluating cybersecurity visualization, and present components of a visualization system that can be evaluated. However, we did not find prior work that provides guidelines based on human factors

engineering concepts and principles to cybersecurity visualization design. We present guidelines drawn from the literature based on our study and analysis to help cybersecurity visualization designers address some human factors challenges in the context of interface design.

Carry out work domain analyses. Cybersecurity visualization designers should first analyze the work domain before they analyze what cyber security defenders do, or what cyber security defenders know. This is because it is impossible to comprehend the behavior of cyber defenders without concurrently comprehending the environment in which they operate. Work domain analyses will provide understanding of functions and constraints of the work domain. Constraints may be identified by using multiple perspectives such as published accounts of standard practices, the opinion of various cyber security experts, field observation within the cyber security domain, laboratory experiments, and experiments with synthetic task environments.

Represent affordances of the work domain in a way that is compatible with human visual capabilities. Visualization is only successful to the extent that it encodes information such that human eyes can discern and human brains can understand [52]. Whereas a good representation can make a problem easier to solve, a bad representation can make the problem much more difficult to solve [51]. The goal of EID should be to uncover the affordances in the cyber security domain, and represent them in a way that is compatible with human visual capabilities. Affordances should be represented as information accessible in the visualization . The visualization should match both the cybersecurity work demands and the perceptual and cognitive skills of cyber defenders. It should allow cyber defenders to execute input directly, that is, the perception-action loop should be kept intact [28]. That way the visualization will serve as an "external model" of the structure of the cyber security work domain [1].

Support the level of cognitive control at which cyber defenders choose to perform. The visualization should support the full range of activities that cyber defenders must engage in, without making the task more complex in any manner. It should make processes more transparent and observable to cyber defenders, so that, as much as possible, they can use lower level perceptual processing rather than higher level analytical processing. This ensures that cybersecurity defenders are free from cognitive overload and only heavily rely on analytical during anomalous events. In their work domain, cyber defenders deal with three events: familiar, unfamiliar but anticipated, and unfamiliar and unanticipated. EID deals with all three event types, thus offering cyber defenders the most appropriate support in any situation. Tasks that involve continuous space-time signals, whereby cybersecurity defenders execute control input by manipulating (e.g., pointing, clicking, dragging, swiping, dropping) objects on the visual interface, support skill-based behavior. Rule-based behavior is supported by a consistent one-to one mapping between the cybersecurity work domain constraints and the perceptual information on the interface, whereby experienced cybersecurity defenders are able to see current system states and manipulate objects in the visualization, instead of reasoning about them. Knowledge based behavior is supported by the use of abstraction hierarchy to remove the cognitive load from the cybersecurity operator and giving them an externalized mental model of the entire work domain.

Provide interface resources that allow cyber defenders to navigate through the work domain. Designers should take advantage of the human visual system's ability to do preattentive processing by seeking to visually encode information pre-attentively. The visualization should present information in a manner that allows the cybersecurity defenders to understand the situation effortlessly. Also, visualization designers should apply gestalt principles to their designs to enable cyber defenders identify patterns that matter, speedily and efficiently. In their work, cyber defenders integrate data across multiple screens, within individual screens, and within a display on a screen. The visualization needs to support the effective distribution of cyber defender attention. Visual momentum may be increased by spatially dedicating controls and displays, fixed format data replacement technique, functional overlap, and the long shot design technique [43].

4 CONCLUSIONS

The design of effective visualizations is dependent upon very specific interactions between the cybersecurity work domain, the visualization, and the cyber defender. Consequently, visualizations must be tailored to simultaneously match both the specific work demands and the powerful perceptual skills of the cyber defender. In this paper, we used an ecological interface design approach to present human factors-based principles of interface design for cybersecurity visualization. Principles are abundant in the literature, but must be targeted to tasks associated with cyber defense.

REFERENCES

[1] Bennett, K. B. (2014, September). Veils: an ecological interface for computer network defense. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 58, No. 1, pp.11 1233-1237). Sage CA: Los Angeles, CA: SAGE Publications.*

[2] Fink, G. A., North, C. L., Endert, A., & Rose, S. (2009) Visualizing cyber security: Usable workspaces. In Visualization for Cyber Security *VizSec 2009. 6th International Workshop on 1 (pp. 45-56). IEEE.*

[3] Chen, Y. V., Qian, Z. C., & Lei, W. T. (2016, June). Designing a Situational Awareness Information Display: Adopting an Affordance-Based Framework to Amplify User Experience in Environmental Interaction Design. *In Informatics (Vol. 3, No. 2, p. 6). Multidisciplinary Digital Publishing Institute.*

[4] Nuamah, J., & Seong. Y. (2017). Human Machine Interface in the Internet of Things (IoT). *A paperpresented at the 2017 IEEE System of Systems Engineering 2017 Conference, Waikoloa, Hawaii, USA.*

[5] Robertson, G., Czerwinski, M., Fisher, D., & Lee, B. (2009). Selected human factors issues in information visualization. . *Reviews of human factors and ergonomics, 5(1), 41-81*

[6] Gutzwiller, R. S., Fugate, S., Sawyer, B. D., & Hancock, P. A. (2015, September). The human factors of cyber network defense. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 59, No. 1, pp. 322-326). SAGE Publications.*

[7] Wiederhold, B. K. (2014). The role of psychology in enhancing cybersecurity. *Cyberpsychology, Behavior, and Social Networking, 1 17, 131Ð132.*

[8] Proctor, R. W., & Chen, J. (2015). The role of human factors/ergonomics in the science of security: decision making and action selection in cyberspace *Human factors, 57(5), 721-727.*

[9] Best, D. M., Endert, A., & Kidwell, D. (2014, November). 7 key challenges for visualization in cyber network defense. *In Proceedings of the Eleventh Workshop on Visualization for Cyber Security (pp. 33-40). ACM.*

[10] Bass, T. (2000). Intrusion detection systems and multisensor data fusion. *Communications of the ACM, 43(4), 99-105.*

[11] Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 37(1),*

32-64.
[12] Vieane, A., Funke, G., Gutzwiller, R., Mancuso, V., Sawyer, B., & Wickens, C. (2016, September). Addressing Human Factors Gaps in Cyber Defense. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 60, No. 1, pp. 770-773). Sage CA: Los Angeles, CA: SAGE Publications.*
[13] Evans, J. S. B., & Stanovich, K. E. (2013). Dual-process theories of higher cognition: Advancing the debate. *Perspectives on psychological science, 8(3), 223-241.*
[14] Patterson, R. E. (2017). Intuitive Cognition and Models of Human and Automation Interaction. *Human Factors, 59(1), 101-115.*
[15] Epstein, S. (2008). Intuition from the perspective of cognitive-experiential self-theory. *Intuition in judgment and decision making, 23, 37.*
[16] Hogarth, R. M. (2002). Deciding analytically or trusting your intuition? The advantages and disadvantages of analytic and intuitive thought. *In Epstein, S. (2008). Intuition from the perspective of cognitive-experiential self-theory.*
[17] Wickens, C. D., Hollands, J. G., Banbury, S., & Parasuraman, R. (2015). *Engineering psychology & human performance. Psychology Press.*
[18] Wickens, C. D., Lee, J.D., Liu, Y., & Becker, S. (2004). *An Introduction to Human Factors Engineering.*
[19] Ware, C. (2012). *Information visualization: Perception for Design. Elsevier.*
[20] Healey, C., & Enns, J. (2012). Attention and visual memory in visualization and computer graphics. *IEEE transactions on visualization and computer graphics, 18(7), 1170-1188.*
[21] Treisman, A. M., & Gelade, G. (1980). *A feature-integration theory of attention. Cognitive psychology, 12(1), 97-136.*
[22] Wolfe, J. M. (1994). *Guided search 2.0 a revised model of visual search. Psychonomic Bulletin & Review, 1(2), 202-238.*
[23] Rosli, M. H. W., & Cabrera, A. (2015). . *Gestalt principles in multimodal data representation. IEEE computer Graphics and Applications, 35(2), 80-87.*
[24] Koffka, K. (2013). *Principles of Gestalt psychology (Vol. 44). Routledge*
[25] Buja, A., Cook, D., & Swayne, D. F. (1996). *Interactive high-dimensional data visualization. Journal of computational and graphical statistics, 1 5(1), 78-99.*
[26] Pinker, S. (1990). *A theory of graph comprehension. Artificial intelligence and the future of testing, 73-126.*
[27] Garae, J., Ko, R. K., & Chaisiri, S. (2016, August). *UVisP: User-centric visualization of data provenance with gestalt principles. In Trustcom/BigDataSE/I SPA, 2016 IEEE (pp. 1923-1930). IEEE.*
[28] Gibson, J. J. (1977). *Perceiving, acting, and knowing: Toward an ecological psychology. The Theory of Affordances, 67-82.*
[29] Norman, D. A. (1999). *Affordance, conventions, and design. interactions, 6(3), 38-43.*
[30] Hartson, R. (2003). *Cognitive, physical, sensory, and functional affordances in interaction design. Behaviour & Information Technology, 22(5), 315-338.*
[31] Maier, J. R., Fadel, G. M., & Battisto, D. G. (2009). *An affordance-based approach to architectural theory, design, and practice. Design Studies, 30(4), 393-414.*
[32] Chen, Y. V., Qian, Z. C., & Lei, W. T. (2016, June). *Designing a Situational Awareness Information Display: Adopting an Affordance-Based Framework to Amplify User Experience in Environmental Interaction Design. In Informatics (Vol. 3, No. 2, p. 6). Multidisciplinary Digital Publishing Institute.*
[33] Klein, G., & Militello, L. (2001). *Some guidelines for conducting a cognitive task analysis. In Advances in human performance and cognitive engineering research (pp. 163-199). Emerald Group Publishing Limited.*
[34] D'Amico, A., Whitley, K., Tesone, D., O'Brien, B., & Roth, E. (2005, September). *Achieving cyber defense situational awareness: A cognitive task analysis of information assurance analysts. In Proceedings of the human factors and ergonomics society annual meeting (Vol. 49, No. 3, pp. 229-233). Sage CA: Los Angeles, CA: SAGE Publications.*
[35] Mahoney, S., Roth, E., Steinke, K., Pfautz, J., Wu, C., & Farry, M. (2010, September). *A cognitive task analysis for cyber situational awareness. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 54, No. 4, pp. 279-283). SAGE Publications.*
[36] Endsley, M. R. (2016). *Designing for situation awareness: An approach to user-centered design. CRC press.*
[37] McKenna, S., Staheli, D., & Meyer, M. (2015, October). *Unlocking user-centered design methods for building cyber security visualizations. In Visualization for Cyber Security (VizSec), 2015 IEEE Symposium on (pp. 1-8). IEEE.*
[38] Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering.*
[39] Vicente, K. J. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work. CRC Press.*
[40] Bennett, K. B., & Flach, J. M. (2011). *Display and interface design. Subtle Science and Exact Art. CRC Press, Boca Raton.*
[41] Burns, C. M., Kuo, J., & Ng, S. (2003). *Ecological interface design: a new approach for visualizing network management. Computer Networks, 43(3), 369-388.*
[42] Flach, J. M., Tanabe, F., Monta, K., Vicente, K. J., & Rasmussen, J. (1998, October). *An ecological approach to interface design. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 42, No. 3, pp. 295-299). SAGE Publications.*
[43] Bennett, K. B., & Flach, J. M. (2012). *Visual momentum redux. International Journal of Human- Computer Studies, 70(6), 399-414.*
[44] Rasmussen, J. (1983). *Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. IEEE transactions on systems, man, and cybernetics, (3), 257-266.*
[45] Woods, D. D. (1984). *Visual momentum: a concept to improve the cognitive coupling of person and computer. International Journal of Man-Machine Studies, 21(3), 229-244.*
[46] Langton, J. T., & Baker, A. (2013, June). *Information visualization metrics and methods for cyber security evaluation. In Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on (pp. 292-294). IEEE.*
[47] Staheli, D., Yu, T., Crouser, R., Damodaran, S., Nam, K., O'Gwynn, D., McKenna, S., & Harrison, L. (2014). *Visualization evaluation for cybersecurity: trends and future directions. VisSec '14: Proceedings of the Eleventh Workshop on Visualization for Cybersecurity.*
[48] Ware, C. (2012). *Information visualization: perception for design. Elsevier.*
[49] Nesbitt, K. V., & Friedrich, C. (2002). *Applying gestalt principles to animated visualizations of network data. In Information Visualisation, 2002. Proceedings. Sixth International Conference on (pp. 737-743). IEEE.*
[50] Ali, N., & Peebles, D. (2013). *The effect of gestalt laws of perceptual organization on the comprehension of three-variable bar and line graphs. Human Factors: The Journal of the Human Factors and Ergonomics Society, 55(1), 183-203.*
[51] Bennett, K. B. (2017). *Ecological interface design and system safety: One facet of Rasmussen's legacy. Applied ergonomics, 59, 625-636.*
[52] Few, S. (2013). *Data visualization for human perception. The Encyclopedia of Human-Computer Interaction, 2nd Ed.*

A Practical Application for Sentiment Analysis on Social Media Textual Data

Colton Aarts
Department of Computer Science
University of Northern British Columbia
Prince George, BC, Canada
aarts@unbc.ca

Fan Jiang
Department of Computer Science
University of Northern British Columbia
Prince George, BC, Canada
Fan.Jiang@unbc.ca

Liang Chen
Department of Computer Science
University of Northern British Columbia
Prince George, BC, Canada
Liang.Chen@unbc.ca

ABSTRACT

With the amount of data that is available today in textual form, it is essential to be able to extract as much useful information as possible from them. While some textual documents are easy to be understood, other textual documents may need extra processes to discover the hidden information within it. For instance, how the author was feeling while writing this piece of text, or what emotions authors are expressing in this piece of text. The idea of discovering what emotions are expressed in a textual document is known as sentiment analysis. The interest in sentiment analysis has been steadily growing in the past decade. Being able to accurately detect and measure the different emotions present in a text has become more and more useful as the availability of online resources has increased. These resources can range from product reviews to social media content. Each of these resources presents their own distinct challenges while still sharing the core techniques and procedures. In this paper, we introduce an application that can detect four distinct emotions from social media posts. We will first outline the techniques we have used as well as our outcomes, then discuss the challenges that we faced, and finally, our proposed solutions for the continuation of this project.

KEYWORDS

Sentiment Analysis, Big Data, Social Media Analysis, Emotion Detection

ACM Reference Format:
Colton Aarts, Fan Jiang, and Liang Chen. 2020. A Practical Application for Sentiment Analysis on Social Media Textual Data. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3410566.3410594

1 INTRODUCTION

It is widely accepted that the foundation of sentiment analysis was done in these following three research works: [2, 18, 21]. Researchers in [18], study movie reviews. In their work, they try to determine if reviews of movies are positive or negative. They compare different approaches and techniques, namely support vector machines, Naive Bayes classifiers, and maximum entropy classifiers. On the other hand, reserachers in [21], try to achieve a similar goal with different techniques. They used part of speech tagging and individual word weights to produce positive or negative labels. Finally, in [2], researchers are no longer satisfied with simple positive/negative classification (binary classification). They tried to use a range based rating system to determine how positive or how negative those reviews are. They test several different methods in their work, also arise several problems of non-binary labels.

As the need for sentiment analysis continued to grow, it attracts many researchers to focus on it. Researchers nowadays are no longer satisfied with simple positive/negative binary sentiment classification. To expand the number of domains and the range of emotions that capable of being detected, we need to create annotated training and testing collections. One of such corpora is studied in [25]. This paper details the creation of a large collection of annotated text that contains a vast amount of information manually extracted from large text pieces. The authors outline the process in which they obtained the labels and explained what the different fields mean in their corpus. This corpus allowed the application of sentiment analysis to a more diverse domain as it was no longer dependent on online reviews. Furthermore, researchers in [17] proposed a system of collecting a corpus from Twitter that contains positive, negative, and neutral tweets. These tweets are labeled positive or negative based on the presence of different emotions. The objective corpus was collected from several news reporting accounts. In [14], researchers first proposed four different emotions labels for textual content, they are: anger, fear, joy, and sadness. This work allows us to classify textual content into more distinct classes, which is much accurate than the binary label system (positive and negative).

Moreover, there are a variety of other different techniques that can be used for textual information classification. In [7], researchers propose a lexical system to classify tweets. In [26], researchers give a review of the different deep learning and neural networks approaches that can be applied to sentiment analysis. Researchers in [8] use both k-nearest neighbors and support vector machines to create two approaches for the labeling of tweets. Last but not least, ensemble classifiers are used in [16]. It uses a deep learning technique in conjunction with surface-level techniques to create an ensemble to classify tweets.

In this paper, we propose an application that can perform sentiment analysis on Twitter text data. Our application creates an

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06.
https://doi.org/10.1145/3410566.3410594

ensemble classifier built from a support vector machine, a lexical classifier, and a Naive Bayes classifier.

The remainder of this paper is organized as follows. The next section will cover the background studies, in which we will introduce some important existing theories and methods that can be applied to sentiment analysis. Section 3 will present our social media text sentiment analysis application by walking through examples. In Section 4, we present the outcome of the application and evaluation of this outcome. In the end, a conclusion will be presented.

2 BACKGROUNDS AND RELATED WORKS

In general, the process of sentiment analysis can be divided into two steps: preprocessing and classification. Each of them can both be further divided into detailed steps. The step of preprocessing transform a document or textual content from whichever the language/style it was written into an acceptable format for future process. In other words, this step determines what are the important content in the original document, and removes all necessary parts for future operation. After preprocessing, documents should be in a form that is easy for algorithms to work with. For instantce, this new form can be in numerical representation that appears as a matrix. After that, the matrix will then be given over to the classification step, where the transformed document will be processed and labeled.

2.1 Preprocessing

In our application, several techniques are used for preprocessing, and they are: stop word removal, stemming, conversion to a Term-Frequency Inverse Document Frequency (TF-IDF) matrix, and then dimension reduction using Singular Value Decomposition (SVD).

2.1.1 Stop Word Removal. Stop words are words that have been identified as words that are common in regular speech and that their presence in a sentence does not influence the content or sentiment of the sentence. The removal of these words helps reduce the overall complexity of the classifier as well as improving the recall of the classifier [24]. The list of stop words that we are using was obtained from [19]. This is a list of English words that occur in a large number of sentences, and they have been determined to have no effect on the overall contents of the sentence. The process of removing stop words is fairly simple. The stop words are held in a set. Each document is split into single words. These words are then checked against the set of stop words. If a word is found to be in the set, it is removed from the list of words that belongs to the document. The removal of stop words helps control the overall dimensions of the search space for the classifier. This, in turn, will make the classifier more efficient and more accurate.

2.1.2 Stemming. Stemming a word is the process that is used to return the word to its root or stem word. An example of this would be to take the word "running" and to change it to the word "run". The steaming process that is used in this application is from the NLTK package [11]. This package calls a method from the Wordnet package [9]. This method works by utilizing two methods. The first of which is that Wordnet has a list of endings that can be attached to words. Each word passed into the method is analyzed to determine if it contains one of the endings. If it does, then ending is removed and reconstruction rules are utilized to return a word that is close to the intended root word. The second method is a list of possible exceptions to the first list. Wordnet strives to utilize these two rules in an intelligent fashion to stem the supplied words in a logical fashion [9].

2.1.3 Term-Frequency Inverse-Document Frequency. Converting the text into a matrix representation is a standard way to represent the material to the computer. One of the common matrix representations used is the TF-IDF matrix. This matrix represents the occurrence of each word in a document against how often the word occurs across all documents. Words that occur in almost all documents are given lower values as they are deemed to be less important than words that occur in a select few documents. We are using scikit-learn's TF-IDF Vectorizor. This vectorizor will create a matrix where each row is a term, and each column represents the different documents [19]. The equation used to calculate each term-document pair is as follows:

$$tfidf(t,d) = tf(t,d) \times idf(t) \tag{1}$$

Where t is the term, d is the document, $tf(t,d)$ is how many time the term t occurs in document d and

$$idf(t) = log(\frac{1+n}{1+df(t)}) + 1 \tag{2}$$

Where n is the total number of documents and $df(t)$ is the number of documents that contain the term t. The ones that are added to the numerator and denominator represent an additional imaginary document that contains all terms exactly once. This is done to prevent division by zero. The one that is added at the end of the equation is used to make sure that terms that occur in every document and not entirely ignored. This is important in our context because we have already removed the stop words at this point, and we want to make sure that all the remaining words are at least available to the classifiers for consideration. The creation of a TF-IDF matrix helps the classifier understand which words are potentially more important than others. However, one of the problems that can arise from using this matrix is that there can be thousands of different words in a corpus. This makes the dimensions of the TF-IDF matrix too large to use by itself. To handle this problem, it is common to implement dimension reduction techniques like SVD.

2.1.4 Singular Value Decomposition. SVD is the process of reducing the number of dimensions that are present in a matrix. This process has been shown to combine words that have been used in similar contexts into a single value [23]. Reducing the number of dimensions (or the rank of the matrix) is useful for sentiment analysis because, in even a small number of sample documents, the rank of the matrix can be incredibly large. This can lead to long processing times and the possibility of losing accuracy to unneeded information.

The process of reducing the rank of a matrix using SVD utilizes the fact that a matrix C of dimensions $M \times N$ can be represented as

$$C = U\Sigma V^T \tag{3}$$

Where U is the $M \times M$ matrix whose columns are the orthogonal eigenvectors of CC^T, and V is the $N \times N$ matrix whose columns are the orthogonal eigenvectors of C^TC [23]. The matrix Σ is the $M \times N$ matrix comprised of the singular values. Singular values are the

ordered square roots of the eigenvalues from the matrix CC^T. In order to reduce the rank of the final output matrix, a person selects a number of the singular values k to keep. You then dispose of all values of Σ that are lower than k and you construct the final matrix $C_k = U\Sigma_k V^T$ [23]. The value of k is important. It is common to choose a value in the hundreds. For our application, we have chosen 800 for this value. This gives us a reasonable number of dimensions while still retaining a good amount of the original information.

2.2 Classification

After the text has been through the preprocessing stage, it is ready to be given to the classifier. The classifier will take in the text and will return a single value determining what sentiment the text is expressing. In our application, we have created an ensemble classifier from an SVM, a lexical classifier, and a Naive Bayes classifier.

2.2.1 Ensemble Classifiers. Ensemble classifiers are a group of classifiers that are used together to classify a piece of text. The reason to use an ensemble is that you are hoping that the different classifiers will cover the individual weaknesses that they could exhibit alone [5]. Authors have found that using ensemble classifiers improves the performance of the classifier when compared to the component classifiers [5, 10, 20]. What component classifiers of the ensemble can be a variety of different classifiers. In [10], the authors compared Naive Bayes, Support Vector Machines, and Maximum Entropy classifiers. The authors of [20] used a Naive Bayes, Maximum Entropy, and a knowledge-based tool. Their knowledge-based classifier is a lexical classifier that takes into consideration modifying words. These modifying words are words that could influence the degree or amount of emotion being expressed. Some examples of these words would be "very", "quite", and "not". In [5], the ensemble classifier is made from a Random Forest, a Naive Bayes, a Support Vector Machine, and a Logistic Regression classifier. There are two main ways to determine the final label when using an ensemble classifier. One way is to take the majority vote from the component classifiers. The other way is to take the average of the probabilities supplied from the components [5]. For our application, we are using a Naive Bayes, a Support Vector Machine, and a Lexical classifier and taking the majority vote from the results. In the case of a nonbinary set of labels, if there is no agreement amongst the classifiers, we will give the document a label of 0.

2.2.2 Support Vector Machines. SVMs are a classifier that is generally used to classify two distinct groups. They do this by constructing a hyperplane that separates the two groups with a maximized margin [4]. This margin is the distance from a point belonging to one of the classes to the plane. For a hyperplane to be considered valuable, it has to maximize the size of this margin for points in both classes [4]. In order to classify multiple different classes, SVMs use one of two different approaches. The first is a one-against-one approach. This approach constructs a classifier for each pair of classes. This means that in the case of n classes it will construct $\frac{n\times(n-1)}{2}$. The other option for multiclass classification is a one-against-all approach. In this approach, a classifier is constructed that compares a class against all other classes. This means that there will only be n classes created. Both of these strategies are valid, and choosing the correct one greatly depends on the problem at hand [13]. The implementation that we are using in this application is from the scikit-learn Python package [19]. For its' support vector classifier (SVC), the one-against-one approach is used [19].

2.2.3 Lexical Classifier. Lexical classifiers are one of the simpler classifier types. They work from a supplied list of emotional words. These words need to be labeled with the corresponding emotions as well as an associated intensity. Using this list of words, the classifier will determine the emotion of a document according to the presence and number of the words that occur in the list. There are different ways to determine the final label for a document. One approach is to sum all the different emotional weights and then return the emotion that has the largest score. One of the advantages of lexical classifiers is that they do not need any training data [12]. As long as they are supplied with a domain-relevant lexicon, they can classify text. We obtained our lexicon of emotional words from [15]. This lexicon contains 5814 words with associated emotions and weights. The emotions that are paired with these words match the emotions that we are hoping to detect. These emotions are anger, joy, sadness, and fear.

2.2.4 Naive Bayes. Naive Bayes classifiers work under the assumption that there is conditional independence between every pair of features [19]. For document classification, this means that all the words are independent of the words that occur around them. In a bag-of-words model, this assumption has already been made, so Naive Bayes is a good choice. There are many different types of Naive Bayes classifiers to choose from. One of the most popular is the Complement Naive Bayes [19]. It has been shown to outperform the other popular Naive Bayes classifiers while still be fast and relatively easy to implement [19]. This form of Naive Bayes classifier works by implementing the standard multi-nominal Naive Bayes as well as including information regarding the complement of each class [19, 22]. It has been shown that the Complement Naive Bayes classifier works better than the Multi-nominal classifier, especially when dealing with imbalanced data sets [22].

3 OUR PROPOSED TECHNIQUE

Our application takes in two parameters. One is a training file, and the other is the testing file. In this section, we will use the following sentence label pairs for training in table 1:

Keep in mind that this is a small data set only for display purposes. The real data set we used for training is much larger than this.

To show how our application works, let's use the following two sentences in table 2 as target.

The labels for these two sentences should end up being JOY and ANGER, respectively. The training file needs to have each sentence or document on its own line with the corresponding label separated by a tilde. This also means that there cannot be any tildes in the documents. The testing file should have each document on its own line as well.

When two files are supplied to our application, the process will look as follows. The application would take the training file and work with it first. In our example, it would take the training sentences and separate the documents from the labels. It would then take the documents and begin to preprocessing them. This would mean that it would stem all the words and remove any words that

Table 1: Training Sentences and Their Emotion Labels

SID	Training Sentences	Emotion Labels
1	I am so happy and excited, I won the lottery!	JOY
2	Today was a good day.	JOY
3	I went shopping today and the grocery store was so busy. There were people everywhere and I was getting annoyed.	ANGER
4	Ahhh I am terrified of thunderstorms.	FEAR
5	All this rain is really depressing.	SADNESS
6	If this traffic doesn't stop soon I am going to get upset.	ANGER
7	Listening to the new can be terrifying sometimes.	FEAR
8	My car broke down today, I almost started crying. I am going to miss that car.	SADNESS

Table 2: Target Sentences for Emotion Detection

SID	Target Sentences	Emotion Labels
1	This is so exciting! I am doing to Disney land!	(Unknown)
2	I was driving today and this person cut me off. I was so pissed off and upset.	(Unknown)

Table 3: Training Sentences and Their Emotion Labels (After Removing Stop Words)

SID	Training Sentences (After Removing Stop Words)	Emotion Labels
1	happi excit , won lotteri !	JOY
2	today good day .	JOY
3	went shop today groceri store busi . people annoy .	ANGER
4	ahh terrifi thunderstorm .	FEAR
5	rain realli depress hope stop soon .	SADNESS
6	traffic doesn't stop soon upset	ANGER
7	listen ne terrifi sometime .	FEAR
8	car broke today , start . miss car .	SADNESS

are in the stop words set. The outcome after the preprocessing is in Table 3.

After this, these documents would be transformed into a TF-IDF matrix. This sparse matrix would have dimensions 8 x 35. After this conversion, the TF-IDF matrix would be transformed into a smaller matrix. In our case, with this example, the matrix is already small, but we will run it through the SVD process for illustration purposes. We will choose the final size to be 8. The size of the final SVD matrix cannot be larger than the number of documents and

Table 4: Target Sentences for Emotion Detection (After Removing Stop Words)

SID	Target Sentences (After Removing Stop Words)	Emotion Labels
1	excit ! disney land !	(Unknown)
2	drive today person cut . piss upset .	(Unknown)

Table 5: Target Sentences and the Analyzed Emotion Label

SID	Target Sentences	Emotion Labels
1	This is so exciting! I am doing to Disney land!	JOY
2	I was driving today and this person cut me off. I was so pissed off and upset.	ANGER

must be smaller than the number of words. This will result in a dense matrix that is 8×8. This new smaller matrix is now used to train the SVM, while the TF-IDF matrix is used for the Naive Bayes classifier. The lexical classifier is not trained as its labels are based on the word value pairs that it is supplied within its lexicon. After the classifiers are trained, then the testing data is put through the same steps. The difference is that the documents are appended to the TF-IDF matrix. This means that new words are ignored, and the only words that are considered are words that were in the training data. The stemming and removal of the stops word from the testing sentences would result in Table 4

Putting these through the TF-IDF matrix that was created from the training data results in a sparse matrix with only 6 values. These are at the words "excit", and "!" for the first sentence and at "upset", "today", and ",". Putting this matrix thought the same SVD process as the training data results in a dense 2 x 8 matrix. This matrix is then given to the SVD, while the TF-IDF matrix is given to the Naive Bayes classifier. The resulting labels are "JOY" and "ANGER" from both classifiers. The lexical classifier results in the same labels, as well as the first sentence, ends with values of 0.172 for anger, 0.132 for fear, 0.0 for sadness, and 0,697 for joy. For the second sentence, the lexical classifier gets the values 1.818 for anger, 0.484 for fear, 0.984 for sadness, and 0.0 for joy. Taking the maximum values returns the final labels of "JOY" and "ANGER".

4 EVALUATION AND ANALYSIS

4.1 Performance of Our Application

The data set that we chose to use to train and test our classifier is made available by [14]. It has 3173 annotated tweets in it. Out of these, there are 997 that are labelled as fear, 727 that are labelled as joy, 750 that are labelled as anger, and 699 that are labelled as sadness. We wanted to test our classifier against the fairly balanced set of all emotions as well as in the unbalanced situations of the binary classification of the four different emotions. For testing and training purposes, we spilled the tweets 70:30 giving us 2219 tweets for training and 954 for testing. The total number of tweets from each class in the training and testing sets was determined randomly. For the fear binary classifier, there are 592 fear tweets in the training

Table 6: Results of Multi Class Labeling

	precision	recall	f1-score
fear	0.84	0.77	0.80
joy	0.84	0.75	0.80
anger	0.48	0.94	0.64
sadness	0.89	0.59	0.71

Table 7: Fear

	precision	recall	f1-score
not fear	0.73	0.98	0.84
fear	0.95	0.52	0.67

Table 8: Joy

	precision	recall	f1-score
not joy	0.89	0.98	0.93
joy	0.91	0.58	0.70

Table 9: Anger

	precision	recall	f1-score
not anger	0.99	0.97	0.98
anger	0.68	0.80	0.74

data and 405 in the testing. For joy, there are 510 in the training and 217 in the testing. There are 684 in the training and 66 in the testing for the anger binary classifier. Finally, there are 433 training examples and 266 testing examples for the sadness classifier. The remaining tweets in all the training and testing sets for the binary classifiers are tweets that are labeled as to not contain the emotion for that classifier. The multi-emotion classifier had the same rations. This means that its training set had 592 fear, 510 joy, 684 anger, and 433 sadness tweets in it. The testing set had the remaining tweets. The evaluation criteria we are using is measuring precision, recall, and F1-score or F-Score. Precision is defined as:

$$precision = \frac{TP}{TP + FP} \tag{4}$$

Where TP is the true positives and FP is the false positives. Recall is defined as:

$$recall = \frac{TP}{TP + FN} \tag{5}$$

Where FN is the false negatives. F1-Score is the harmonic mean of recall and precision. It is defined as:

$$F1 - Score = 2 \times \frac{precision \times recall}{precision + recall} \tag{6}$$

The results for the binary classification of fear, joy, anger, and sadness can be seen in tables 7, 8, 9, 10, respectively.

Overall the performance of the lexical classifier is lacking. However, the lexical classifier does offer a benefit that is not seen in the tables. The other two classifiers will suffer when exposed to information that is not in their training set. It is our hope the lexical classifier will start to offer help in this scenario. It appears that,

Table 10: Sadness

	precision	recall	f1-score
not sadness	0.81	98	0.89
sadness	0.89	0.40	0.55

in most cases, the performance of the ensemble classifier overall is very similar to the performance of the SVM classifier. However, there are some cases where there are distinct differences. In table 6 for the fear label, we loose 0.09 recall we gain the same in precision. The difference in the joy label is even more impressive. We lose only 0.01 precision to gain 0.1 recall. In addition to this, all the classifiers perform worse on the anger labels. This could be attributed to unbalanced training and testing amounts. The low recall from the binary classifiers could be caused by a relative imbalance of labelled data. In those classifications, there is a larger volume of labelled data that falls under the "not" class when compared to the target class. However, the performance in the multi-label case is adequate overall. The notable exception of this is with the anger label.

4.2 Compare with Other Existing Work

The authors in [3] propose a model for the automatic detection of emotions from posts on Twitter. In order to create the model that they are proposing, they first needed to collect a large number of tweets for the different emotions that they were planning on classifying. They collected three different sets of Tweets based on the three different definitions of emotions: Ekman, Plutchick, and POMS. After filtering they collect < 535,000 tweets for Ekman, < 798 000 for Plutchik and < 6 500 000 for POMS. However, this work used a different emotion model compare with our proposed application. The closest model to the emotions that we are classifying is the Ekman emotion set. In this set, they are classifying tweets into one of six emotions. These being anger, disgust, fear, joy, sadness, and surprise. With this dataset collected, the authors were able to train their classifier. Their goal was to create a deep learning classifier that was able to classify a tweet for all three categories at once. In addition to this goal, the authors also reported the performance of their neural network for each category individually. The best F1-score that they achieved for their classifier was 60.7 for a multi-class classifier averaged across all the emotional labels. When compared to our multi-class classifier, we can see that ours performs in a comparable manner with an averaged F1-score of 0.73.

In [1] the authors utilize the BERT framework to create an application that is capable of predicting the positive or negative sentiment of a Tweet from Twitter. The BERT framework is defined in [6]. The authors of [6] created a pre-trained deep, bidirectional sentiment classifier that was trained on a large amount of data. This classifier performed well for a number of different NLP tasks. BERT is different from previous approaches as it is trained in a bidirectional manner. This means that it learns to be able to predict a language from both the right-hand side as well as the left. This helps to improve its performance and its capabilities. In [1], the authors take the BERT framework and apply it to sentiment analysis on Twitter. To do this, they collect a large number of Tweets (> 2 million) and use these to pre-train BERT. After this pre-training,

they change the BERT framework slightly to utilize different neural network structures for the final classification. The final results that the authors found was that their approach could get an F1-score of 72.61. While it is not possible to directly compare this result to our application, our application does perform in a comparable fashion. In the case of multi-label classification, anger is significantly lower at 64, and sadness is slightly lower at 71 while both fear and anger are higher at 80. For the cases of the binary classifiers, joy, fear and sadness are all lower while anger is higher. If we had access to a dataset that was large enough to train BERT on and the resources needed to continue the fine-tuning to emotional detection, it would be interesting to see what results could be attained by using this method.

5 CONCLUSION AND FUTURE WORKS

From official news to personal tweets, textual data is everywhere in our life. Therefore, it is getting more and more important to be able to analyze textual data. Sentiment analysis, as one of the most important types of textual data analysis, can detect emotions from textual data. Initially, sentiment analysis can only label textual data in a binary way. That is, it can only determine if a piece of text expressed positive or negative emotion. However, researchers nowadays are no longer satisfied with this simple binary labeling. We need applications that can detect more specific emotions like fear, joy, sadness, etc. In this paper, we propose our application that can analyze textual documents and determine the emotion that was expressed by that document. We apply our application on Twitter textual data that we collected using Twitter API. The evaluation result shows the practicality of our proposed application.

The future of this application holds many promising avenues of research. Working to improve the performance of lexical classifier is one of the most important ones. If the precision and recall of the lexical classifier can be improved, we predict that the overall performance and stability of the classifier will increase a great deal as well. Another area of research that holds interest to us is to apply this application to other areas of sentiment analysis. Currently, we have only tested it on a single corpus. It would be interesting to see how it performs on a different corpus with different labels. We are hoping to use it as a solid base for future research projects. Furthermore, one of the results that can be seen from the different classifiers is the drastic drop in performance when comparing the multi-class labeling to the binary classifications. It would be of interest to investigate further to see if there is a definitive cause for this performance drop and to see if there is a solution. Last but not least, we plan to explore more emotions, for instance, sarcasm, irony, cynical, etc.

6 ACKNOWLEDGMENTS

The work is partially supported by NSERC Discovery Development Grant #2018-00021.

REFERENCES

[1] Azzouza, N., Akli-Astouati, K., Ibrahim, R.: TwitterBERT: Framework for Twitter Sentiment Analysis Based on Pre-trained Language Model Representations. In: 4th International Conference of Reliable Information and Communication Technology (IRICT 2019), pp. 428–437. Springer, Johor-Malaysia, Malaysia (2019)
[2] Bo P., Lillian L.: Seeing Stars: Exploiting Class Relationships For Sentiment Categorization With Respect To Rating Scales. In: the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 115–124. Association for Computational Linguistics, Ann Arbor, Michigan (2005)
[3] Colneriĉ, N., Demsar, J.: Emotion recognition on twitter: Comparative study and training a unison model. IEEE transactions on affective computing. (2018)
[4] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning **20**(3), 273–297 (1995)
[5] Da Silva, N., Hruschka, E., Hruschka Jr, E.: Tweet sentiment analysis with classifier ensembles. Decision Support Systems **66**, 170–179 (2014)
[6] Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2019), pp. 428–437. Minneapolis, USA (2019)
[7] Hassan S., Yulan H., Miriam F., Harith A.: Contextual semantics for sentiment analysis of Twitter. Information Processing and Management **52**(1), 5–19 (2016)
[8] Huq, R., Ali, A., Rahman, A.: Sentiment analysis on Twitter data using KNN and SVM. International Journal of Advanced Computer Science and Applications **8**(6), 19–25 (2017)
[9] Ingo F., Kurt H.: wordnet: WordNet Interface. R package version 0.1-14, https://CRAN.R-project.org/package=wordnet. Last accessed May 2020
[10] Kanakaraj, M., Guddeti, R.: Performance analysis of Ensemble methods on Twitter sentiment analysis using NLP techniques. In: 9th International Conference on Semantic Computing (ICSC 2015), pp. 169–170. IEEE, California, USA (2015)
[11] Loper, E., Bird, S.: NLTK: the natural language toolkit. In: the ACL Interactive Poster and Demonstration Sessions, pp. 214–217. Association for Computational Linguistics, Barcelona, Spain (2004)
[12] Melville, P., Gryc, W., Lawrence, R.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009), pp. 1275–1284. ACM, Paris, France (2009)
[13] Milgram, J., Cheriet, M., Sabourin, R.: "One against one" or "one against all": Which one is better for handwriting recognition with SVMs?. In: 10th International Workshop on Frontiers in Handwriting Recognition, Suvisoft, La Baule, France (2006)
[14] Mohammad, S., Kiritchenko, S.: Understanding emotions: A dataset of tweets to study interactions between affect categories. In: the 11th International Conference on Language Resources and Evaluation. ELRA, Miyazaki, Japan (2018)
[15] Mohammad, S., Turney, P.: Crowdsourcing a Word-Emotion Association Lexicon. Computational Intelligence **29**(3), 436–465 (2013)
[16] Oscar A., Ignacio C., J. F.-S., Carlos I.: Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Systems with Applications **77**, 236–246 (2017)
[17] Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: the 7th International Conference on Language Resources and Evaluation, pp. 1320–1326. ELRA, Valletta, Malta (2010)
[18] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp. 79–86. Association for Computational Linguistics, Philadelphia, PA, USA (2002)
[19] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
[20] Perikos, I., Hatzilygeroudis, I.: Recognizing emotions in text using ensemble of classifiers. Engineering Applications of Artificial Intelligence **51**, 191–201 (2016)
[21] Peter D.-T.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. (2002)
[22] Rennie, J., Shih, L., Teevan, J., Karger, D.: Tackling the poor assumptions of naive bayes text classifiers. In: 20th international conference on machine learning, pp. 616–623. AAAI Press, Washington DC, USA (2003)
[23] Schütze, H., Manning, C.-D., Raghavan, P.: Introduction to information retrieval. Cambridge University Press, Cambridge (2008)
[24] Silva, C., Ribeiro, B.: The importance of stop word removal on recall values in text categorization. In: the International Joint Conference on Neural Networks volume 3, pp. 1661–1666. IEEE, Portland, OR, USA (2003)
[25] Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language resources and evaluation **39**(2-3), 165–210 (2005)
[26] Zhang, L., Wang, S., Liu, B.: Deep learning for sentiment analysis: A survey. WIREs Data Mining and Knowledge Discovery **8**(4), e1253 (2018)

Detecting Fake News by Image Analysis

Elio Masciari*
Vincenzo Moscato
Antonio Picariello
Giancarlo Sperlí
elio.masciari@unina.it
vincenzo.moscato@unina.it
antonio.picariello@unina.it
giancarlo.sperli@unina.it
University of Naples "Federico II"
Naples, Italy

ABSTRACT

The uncontrolled growth of fake news creation and dissemination we observed in recent years causes continuous threats to democracy, justice, and public trust. This problem has significantly driven the effort of both academia and industries for developing more accurate fake news detection strategies. Early detection of fake news is crucial, however the availability of information about news propagation is limited. Moreover, it has been shown that people tend to believe more fake news due to their features [10]. In this paper, we present our framework for fake news detection and we discuss in detail an approach based on deep learning that we implemented by using Google Bert features. Our experiments conducted on two well-known and widely used real-world datasets suggest that our method can outperform the state-of-the-art approaches and allows fake news accurate detection, even in the case of limited content information.

CCS CONCEPTS

• **Computing methodologies** → **Visual inspection**; **Supervised learning by classification**.

KEYWORDS

Fake News, Multimedia analysis, Deep Learning, Social Media

ACM Reference Format:
Elio Masciari, Vincenzo Moscato, Antonio Picariello, and Giancarlo Sperlí. 2020. Detecting Fake News by Image Analysis. In *24th International Database Engineering & Applications Symposium (IDEAS 2020), August 12–14, 2020, Seoul, Republic of Korea.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3410566.3410599

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06…$15.00
https://doi.org/10.1145/3410566.3410599

1 INTRODUCTION

Fake news detection dates back long time ago[11] as journalist and scientists fought against misinformation since the beginning of information sharing by traditional media. As a matter of fact, the pervasive use of internet for communication boosted a quicker spread of false information. Indeed, the term *fake news* has grown in popularity in recent years, especially after the 2016 United States elections but there is still no standard definition of fake news [7]. Aside the definition that can be found in literature, one of the most well accepted definition of fake news is the following: *Fake news is a news article that is intentionally and verifiable false and could mislead readers* [2]. There are two key features of this definition: authenticity and intent. First, fake news includes false information that can be verified as such. Second, fake news is created with dishonest intention to mislead consumers[8].

The content of fake news exhibits heterogeneous topics, styles and media platforms, it aims to mystify truth by diverse linguistic styles while insulting true news. Thus, fake news detection on social media poses peculiar challenges due to the inherent nature of social networks that requires both the analysis of their content [6] and their social context[3, 9].

Fake news detection problem can be formalized as a classification task thus requiring features extraction and model construction. With the advent of social network, it has been observed a huge increase in the volume of image data generated in the last decade. The use of image editing software like Adobe Photoshop or GNU Gimp to create forged images is a major concern for internet companies. These images, often used in fraudulent ways, are prime sources of fake news, thus before any reaction take place, we must verify their authenticity. Our goal is to improve the existing approaches defined so far when fake images are intentionally produced to mislead users by mimicking true ones. More in detail, traditional approaches are based on verification by human editors and expert journalists but do not scale to the volume of news content that is generated in online social networks. As a matter of fact, the huge amount of data to be analyzed calls for the development of new computational techniques. It is worth noticing that, such computational techniques, even if the news is detected as fake, require some sort of expert verification before being blocked. In our framework, we perform an accurate pre-processing of news data and then we apply three different approaches. The first approach is based on classical

classification approaches. We also implemented a deep learning approach that leverages neural network features for fake news detection. Finally, for the sake of completeness we implemented some multimedia approaches in order to take into account misleading images. We discuss in this paper the multimedia approach in more detail.

2 OUR FAKE NEWS DETECTION FRAMEWORK

Our framework is based on news flow processing and data management after performing a pre-processing block which execute filtering and aggregation operation over the news content. Moreover, filtered data are processed by two independent modules: the first one performs natural language processing over data while the second one performs a multimedia analysis.

Our framework is based on news flow processing and data management after performing a pre-processing block which execute filtering and aggregation operation over the news content. Moreover, filtered data are processed by two independent modules: the first one performs natural language processing over data while the second one performs a multimedia analysis.

The overall process we execute for fake news detection is depicted in Figure 1.

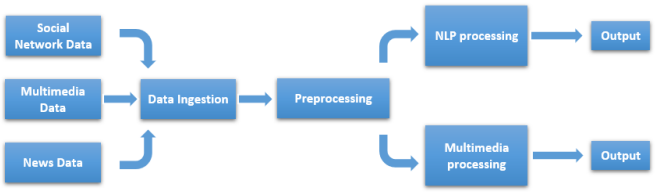


Figure 1: The overall process at a glance

In the following we describe each module in more detail.

Data Ingestion Module. This module take care of data collection tasks. Data can be highly heterogeneous: social network data, multimedia data and news data. We collect the news text and eventual related contents and images.

Pre-processing Module. This component is devoted to the acquisition of the incoming data flow. It performs filtering, data aggregation, data cleaning and enrichment operations.

NLP Processing Module. It performs the crucial task of generating a binary classification of the news articles, i.e., whether they are fake or reliable news. It is split in two sub-modules. The *Machine Learning* module performs classification using an ad-hoc implemented algorithms after an extensive process of feature extraction and selection TF-IDF based (in order to reduce the number of extracted features). The *Deep Learning* module classify data by Google Bert framework after a tuning phase on the vocabulary. It also perform a binary transformation and eventual text padding in order to better analyze the input data.

Multimedia Processing Module. This module is tailored for Fake Image Classification through Deep Learning algorithms, using ELA (Error Level Analysis) and CNN.

Due to space limitation, we discuss in the following only the details of the deep learning module and the obtained results.

2.1 The Deep Learning Module in detail

The Deep Learning Module computes a binary classification on a text datasets of news that will be labelled as 0 if a news is marked as Real, and as 1 if it is marked as Fake. The Deep Learning Module classifies news content using a new language model called *B.E.R.T.* (Bidirectional Encoder Representations from Transformers) developed and released by Google. Prior to describing the algorithm features in detail, we briefly describe the auxiliary tools being used, while in Section 3 we describe the experimental evaluation that lead to our choice on BERT.

Colaboratory. Colab[1] is intended for machine learning education and research, it requires no setup and runs entirely on the cloud. By using Colab it's possible to write and execute code, save and share analytics and it provides access to expensive and powerful computing resources for free by a web interface.

More in detail, Colab's hardware is powered by: Intel(R) Xeon(R) CPU @ 2.00GHz, nVidia T4 16 GB GDDR6 @ 300 GB/sec, 15GB RAM and 350GB storage. This setting is able to speed-up the learning task execution up to 35X and 16X faster in deep learning training compared to a CPU-only server.

Tensor Flow. It is devoted to train and run neural networks for image recognition, word embeddings, recurrent neural networks, and natural language processing. It is a cross-platform tool and runs on CPUs, GPUs, even on mobile and embedded platforms. TensorFlow[1] uses dataflow graphs to represent the computation flow, i.e., these structures describe the data flow through the processing nodes. Each node in the graph represents a mathematical operation, and each connection between nodes is a multidimensional data array called tensor. The TensorFlow Distributed Execution Engine abstracts from the supported devices and provides a high performance core implemented in C++ for the TensorFlow platform. On top there are Python and C++ front ends. The Layers API provides a simple interface for most of the layers used in deep learning models. Finally, higher-level APIs, including Keras, makes training and evaluating distributed models easier.

Keras. It is a high-level neural network API[2], implemented in Python and capable of running on top of TensorFlow. It allows for easy and fast prototyping through: 1) User Friendliness as it offers consistent and simple APIs that minimizes the number of user actions required for common use cases; 2) Modularity as neural layers, cost functions, optimizers, initialization schemes, activation functions and regularization schemes are all standalone modules that can be combined to create new models; 3) Extensibility as new modules are simple to add as new classes and functions.

Google BERT. This tool has been developed in order to allow an easier implementation of two crucial tasks for Natural Language Processing (NLP): Transfer Learning through unsupervised pre-training and Transformer architecture. The idea behind Transfer Learning is to train a model in a given domain on a large text corpus, and then leverage the gathered knowledge to improve the model's

[1]https://research.google.com/colaboratory
[2]https://keras.io

performance in a different domain. In this respect, BERT[3] has been pre-trained on Wikipedia and BooksCorpus. On the opposite side, the Transformer architecture processes all elements simultaneously by linking individual elements through a process known as attention. This mechanism allows a deep parallelization and guarantee higher accuracy across a wide range of tasks[4]. BERT outperforms previous proposed approaches as it is the first unsupervised, fully bidirectional system for NLP pre-training. BERT's model architecture is based on a multi-layer bidirectional Transformer Encoder, an attention mechanism that learns contextual relations between words (or sub-words) in a text. Transformer includes two different mechanisms — an encoder that reads the input text and a decoder that produces a prediction for the task. Since BERT's goal is to generate a language model, only the encoder mechanism has to be properly manipulated. The Encoder's input embedding depicted in Figure 2 and it is composed by: i) token embeddings: it represents the word vector. The first word is the CLS token that is used as a delimiter. It can be used for classification tasks, on the contrary for non-classification tasks, the CLS token can be ignored; ii) segmentation embeddings: it is used to distinguish between two sentences as pre-training can be seen a classification task with two sentences as input; iii) position embeddings: it encodes word ordering.

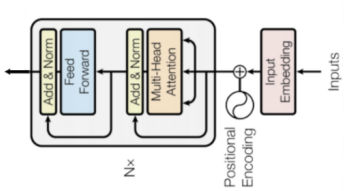


Figure 2: The Econder input

The data flow through the Encoder Architecture is described in what follows: 1) The model represents each token as a vector of *emb_dim* size (dimension of the token embeddings). By assigning one embedding vector for each of the input tokens, we obtain a matrix whose dimensions are *input_length* and *emb_dim* for each input sequence; 2) It then adds positional information (positional encoding). The approach chosen is to add values between [-1,1] using predetermined (non-learned) sinusoidal functions to the token embeddings. Words will be represented slightly differently depending on their position (even for same word). This step builds again a matrix having dimensions *input_length* and *emb_dim*. 3) Data are elaborated by N encoder blocks. Each encoder block is Multi-Head Attention layer that computes *h* different attention values by different weight matrices and then concatenates the results. This step allows the model leverage different representation sub-spaces for different word positions and the use of different filters to create different features maps in a single layer. Its purpose is to find relationships between the input representations and encode them in output. After this step, we obtain a vector of hidden size (768 in BERT Base and 1024 in BERT Large). This vector is used as input on a single-layer neural network classifier to obtain the final output.

[3]https://github.com/google-research/bert
[4]https://towardsdatascience.com/deconstructing-bert-part-2-visualizing-the-inner-workings-of-attention-60a16d86b5c1

After pre-elaboration BERT works in two steps: *pre-training*, i.e., the model is trained on unlabelled data over different tasks and *fine-tuning*, i.e., the BERT model initialized with the pre-trained parameters is fine-tuned using labelled data from the downstream tasks. Each downstream task has separate fine-tuned models, even though they are initialized with the same pre-trained parameters.

Furthermore, BERT is able to build composite data representations to understand language features by Attention mechanism. This task is performed by BertViz, an interactive tool that visualizes attention pattern in BERT from multiple perspectives, i.e., Attention-Head View and Multi-Head Attention View. In Attention-Head View the visualization shows the attention induced by a sample input text. This view visualizes attention as lines connecting the word being updated (left) with the word being attended to (right) as shown in Figure 3.

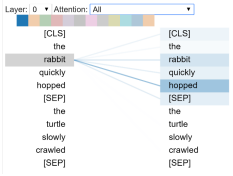


Figure 3: BERT Attention-Head View

Colours encode the attention weight: weights close to one are represented as darker lines, while weights close to zero appear as almost invisible.

Multi-Head Attention in BERT learns multiple attention mechanisms, called heads, which operate in parallel enabling the model to capture a broader range of relationships between words. As the attention heads do not share parameters, each head learns a unique attention pattern.

2.2 The Multimedia module.

This module performs a Fake Image Classification by Deep Learning algorithms, using ELA (Error Level Analysis) and CNN (Convolutinal Neural Networks), with the goal to find if an image has been manipulated or not. Thus, an image related to a news article, will first be submitted to an ELA and then will be labelled as 0 if it is recognised as Real, i.e., it has been not manipulated, and as 1 if it is recognised as Fake, i.e., it has been manipulated. The Multimedia Deep Learning Module has been developed using a Python 3 kernel in a Jupyter. For the implementation, the above describes libraries have been used: Keras, Scikit and Numpy. Moreover, we leveraged the modules described below.

Pandas. It is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming.

Pillow (PIL). It is the Python Imaging Library, a free library for the Python programming language that adds support for accessing and manipulating several different image file formats.

Numpy. It is a library for Python programming language, tailored for large multi-dimensional arrays and matrices manipulation by a huge collection of high-level mathematical functions.

Matplotlib. It is a plotting library for the Python programming language and its numerical mathematics extension NumPy.

2.2.1 Parameter Setting. CNN are complex networks that require many hyper parameters to be set as their values heavily affect the quality of the obtained results. As a matter of fact, the tuning phase requires many tests to be conducted in order to find optimal parameter assignments. In our framework, we manipulated the hyper parameters reported below:

- Architecture-Type and number of hidden layers: the number of hidden layers defines the depth of the network. The depth of the proposed layers has been consistently increased and in general performs better than a shallow network;
- Optimizers: the selected optimizers for investigation are Momentum, RMSProp and Adam. After a deep experimental evaluation we choose Adam;
- Activation Function: the activation function used is ReLU. For binary-classification Sigmoid and Softmax can be used for the last layer. In our framework we choose Sigmoid;
- Dropout Regularization: a regularization technique which avoid overfitting during the training;
- Convolution Layer: there are many parameters that can be changed, however, it is the number of kernels applied to each layer, the height and width of each convolutional kernel and padding;
- Dimensions of pooling matrix: the most commonly used size for pooling is 2x2, i.e., images are half down sampled. A larger pooling matrix size would increase the down sampling rate;
- Number of Epochs: defines the number times that the learning algorithm will work on the entire training dataset. We set this value to 10;
- Batch size: defines the number of samples used, before updating the internal model parameters. Possible values are 16, 32, 64. We found in our experiments that the optimal value is 32.

Figure 4: Real Image

As an example of the system output, we report in Figure 4 and 5 the plots obtained with our system for a Fake and Real image.

3 OUR BENCHMARK

In this section we will describe the fake news detection process for the deep learning module and the datasets we used as a benchmark for our algorithms.

3.1 Dataset Description

FakeNewsNet. This dataset has been built by gathering information from two fact-checking websites to obtain news contents for

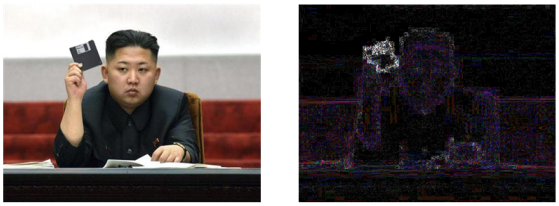


Figure 5: Fake Image

fake news and real news such as PolitiFact and GossipCop. In PolitiFact, journalists and domain experts review the political news and provide fact-checking evaluation results to claim news articles as fake or real. Instead, in GossipCop, entertainment stories, from various media outlets, are evaluated by a rating score on the scale of 0 to 10 as the degree from fake to real. The dataset contains about 900 political news and 20k gossip news and has only two labels: true and false[14]. We decided to analyse only political news as they produce worse consequences in real world than gossip ones. The dataset is well balanced and contains 434 real news and 367 fake news. Most of the news regards the US as it has been evaluated also in LIAR. Fake news topics concern Obama, police, Clinton and Trump while real news topics refer to Trump, Republicans and Obama. Such as the LIAR dataset, it has been added a new column and used the command $df.describe()$ to print out the following statistical information: count 801, mean 1459.217228, std 3141.157565, min 3, 25% 114, 50% 351, 75% 893, max 17377.

The average number of words per articles in Politifact dataset is 1459, which is far longer than the average sentence length in Liar Dataset that is 19 words per articles. Such a statistics suggested us to compare the model performances on datasets with such different features.

Moreover, among the available columns we can access $main-img$ that containis the URL of to the main image in the article. The latter feature allows us to use this dataset also for multimedia analysis. After a preliminary initial check relating to the validity of the URL provided by the dataset, the image file has been downloaded and stored for multimedia analysis.

PHEME Dataset. Among the available labelled datasets containing both real and fake news with related image, we used PHEME Dataset to train the classifier as it contains several news categories from politics to general news. The original dataset is partitioned into nine folders containing breaking news events. It is structured as follows: each event has a directory, with two subfolders, rumours and non-rumours. These two folders contain additional folders named with a news ID and each of these contains two different file: *annotation.json*, which contains information about veracity of the rumour and *structure.json*, which contains information about structure of the conversation. The dataset contains 15k news articles with the main features linked to it along with the URLs of the corresponding files. Each of them has a label which is 0 if the statement and, consequently the image, is *Real*, or *Not Manipulated*, and 1 if the statement, and the image, is *Fake*, or *Manipulated*.

Finally, the dataset is partitioned as follows: i) Training Set: 1779 real image and 2143 fake image; ii) Validation Set]: 771 real image and 895 fake image; iii) Test Set: 771 real image and 895 fake image.

4 EVALUATION

As mentioned in previous section we performed an accurate parameter tuning. In Figure6 we report an excerpt of our setting steps on PHEME dataset (similar results have been obtained on the other datasets). We fixed the following parameter for CNN1: Number of Epoch = 10; Batch size = 32; Learning Rate = 0.001; Pooling matrix = 2x2; Dropout= 0.5; Input Shape = (128,128) and Activation Function= ReLU. We compared the performances on well-established evaluation measure like: Accuracy, Precision, Recall, F1 measure, Area Under Curve (AUC) [5] and the values reported in the obtained confusion matrices for each algorithm, i.e., True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN).

CNN	Accuracy	Precision	Recall	F1	TP	FP	TN	FN	AUC
CNN #1	0.512	0.482	0.437	0.458	302	324	651	389	0.501
CNN #2	0.623	0.568	0.597	0.582	549	418	329	370	0.662
CNN #3	0.745	0.719	0.676	0.697	571	223	598	274	0.741
CNN #4	0.758	0.740	0.770	0.755	598	210	680	178	0.753

Figure 6: Parameter Tuning for Pheme Dataset

In order to try to improve the accuracy we changed for CNN2 the learning rate to 0.001 and the activation function as sigmoid. The results showed a 20% improvement in accuracy. The latter is due to the new optimizer value, which combines the heuristics of both Momentum and RMSProp, and the different function used for the last layer, which performs better in binary-classification. To further improve the results, we implemented CNN3 by adding two additional layers that caused a further accuracy increase. Finally, we used a (3x3) kernel sizes that results in a lower number of weights and higher number of layers that turns out to be a more computationally efficient choice. Hence, we can conclude that 3x3 convolution filters will be a better choice.

In Figure 7, we report the confusion matrix for CNN4 on PHEME dataset.

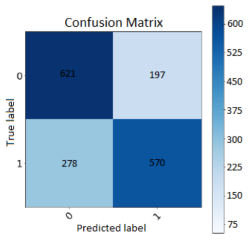


Figure 7: Confusion Matrix for PHEME dataset

We hypothesize that our results are quite better due to a fine hyper parameter tuning we performed, a better pre-processing step and the proper transformation. For the sake of completeness, we report in Figure 8 and Figure 9 the accuracy measures and confusion matrix obtained by CNN4 on Polifact datasets.

5 CONCLUSION AND FUTURE WORK

In this paper, we investigated the problem of fake image detection by deep learning algorithms. We developed a framework that leverages

Model	Accuracy	Precision	Recall	F1	TP	FP	TN	FN	AUC
CNN #4	0.765	0.679	0.705	0.692	220	104	201	92	0.751

Figure 8: Evaluation Measures for Polifact dataset

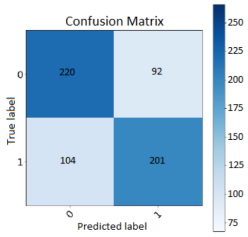


Figure 9: Confusion Matrix for Polifact dataset

CNN for analyzing real-life datasets and the results we obtained are quite encouraging. As future work, we would like to extend our analysis by considering also user profile features and some kind of dynamic analysis of news diffusion mechanism in our fake news detection model.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, Savannah, GA, USA, November 2-4, 2016*, Kimberly Keeton and Timothy Roscoe (Eds.). USENIX Association, 265–283. https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

[2] Hunt Allcott and Matthew Gentzkow. 2017. *Social Media and Fake News in the 2016 Election.* Working Paper 23089. National Bureau of Economic Research. https://doi.org/10.3386/w23089

[3] Nunziato Cassavia, Elio Masciari, Chiara Pulice, and Domenico Saccà. 2017. Discovering User Behavioral Features to Enhance Information Search on Big Data. *TiiS* 7, 2 (2017), 7:1–7:33. https://doi.org/10.1145/2856059

[4] J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.). 2019. *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019.* ACM. https://doi.org/10.1145/3289600

[5] Peter A. Flach and Meelis Kull. 2015. Precision-Recall-Gain Curves: PR Analysis Done Right. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (Eds.). 838–846. http://papers.nips.cc/paper/5867-precision-recall-gain-curves-pr-analysis-done-right

[6] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2017. A Stylometric Inquiry into Hyperpartisan and Fake News. *CoRR* abs/1702.05638 (2017). arXiv:1702.05638 http://arxiv.org/abs/1702.05638

[7] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *CoRR* abs/1708.01967 (2017). arXiv:1708.01967 http://arxiv.org/abs/1708.01967

[8] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *CoRR* abs/1708.01967 (2017). arXiv:1708.01967 http://arxiv.org/abs/1708.01967

[9] Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond News Contents: The Role of Social Context for Fake News Detection, See [4], 312–320. https://doi.org/10.1145/3289600.3290994

[10] Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science* 359, 6380 (2018), 1146–1151. https://doi.org/10.1126/science.aap9559 arXiv:https://science.sciencemag.org/content/359/6380/1146.full.pdf

[11] Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. 2019. Fake News: Fundamental Theories, Detection Strategies and Challenges, See [4], 836–837. https://doi.org/10.1145/3289600.3291382

AN ANDROID-BASED MOBILE PARATRANSIT APPLICATION FOR VULNERABLE ROAD USERS

Kelvin Kwakye†
Industrial & Systems Engineering
North Carolina A&T State University
Greensboro NC USA
kkkwakye@aggies.ncat.edu

Younho Seong
Industrial & Systems Engineering
North Carolina A&T State University
Greensboro NC USA
yseong@ncat.edu

Sun Yi
Mechanical Engineering
North Carolina A&T State University
Greensboro NC USA
syi@ncat.edu

ABSTRACT

Making an optimal travel plan is not an easy task, mostly for vulnerable road users like the elderly and people with mobility disabilities. This optimal travel plan is dependent on the time of day to travel, the route to ply, ways to navigate, and suitable mode of transportation as these vulnerable road users need to reserve paratransit ahead of time. Researchers and mobile app developers are faced with difficulties in incorporating all necessary information that would ease navigation and bus riding for vulnerable road users (that is, making a more assistive system for vulnerable road users). In response to these challenges, this research is designed to develop an interactive android mobile application to ease and significantly encourage vulnerable road users (VRUs) especially the elderly and the disabled to use paratransit to improve their traveling experiences.

This paper assesses the challenges facing the paratransit service. It provides an improvement in areas of reservation, dispatch & routing, and user experience. In response to these challenges, an interactive android mobile application has been developed to ease and significantly encourage vulnerable road users (VRUs) especially older adults and the disabled to use paratransit services and improve their traveling experiences.

CCS CONCEPTS

• Human-centered computing • Ubiquitous and mobile computing • Ubiquitous and mobile devices

ACM Reference format:

Kelvin Kwakye, Younho Seong, and Sun Yi. 2020. An Android-Based Mobile Paratransit Application for Vulnerable Road Users. In *Proceedings of ACM IDEAS conference (IDEAS'20). ACM, New York, NY, USA, 5 pages.* https://doi.org/ 10.1145/3410566.3410596

IDEAS 2020, August 12–14, 2020, Seoul, Republic of Korea

ACM ISBN 978-1-4503-7503-0/20/06 $15.00
https://doi.org/10.1145/3410566.3410596©

KEYWORDS

Paratransit, Vulnerable Road Users, Mobility

1 Introduction

Public transportation systems play an increasingly vital part in the independence of the vulnerable road users in society such as older adults and persons with disabilities or reduced mobility and orientation to move around their communities. The demand for paratransit continues to proliferate in the U.S as a result of its aging population. The existing paratransit models cannot afford to meet the increasing demands of this aging group. Among this aging population comes another minority group who are physically disabled and mobility impaired. Mobility of these individuals can be very challenging since specialized vehicles are required for their movement, which usually comes at a high cost. Studies have shown that these riders reportedly experience poor rides.

According to the United States Government Accountability Office, 73 percent of U.S. transit agencies experienced an increase in Americans with Disabilities Act (ADA) paratransit registrants between 2007 and 2010 [1]. Paratransit provides mobility to about 15 million Americans with cognitive disabilities (including developmental disabilities, traumatic brain injury, stroke, and Alzheimer's) [2]and the growing elderly population who can no longer drive or prefer not to drive. Paratransit is, therefore, one of the modes used by most cities to transport these minority groups. Yet, these vulnerable in society continue to face difficulties despite the availability of mobile assistance platforms or transit navigation apps.

In most cities, paratransit trips must be reserved one to two days in advance. Typically, reservations are made by telephone, although some cities provide web-based reservation systems. In most cities, same-day paratransit trips are unavailable. Passengers are assigned pickup times, typically with 30-minute windows; they must meet the vehicle within five minutes of arrival under penalty of being marked as a "no-show". Trips are provided either door-to-door or to the nearest accessible transit service, depending on the user's evaluation. Day-before reservations and unpredictable travel times relegate paratransit users to travel during a particular time.

2 Literature

Over the past 50 years, a social movement for disability rights has quietly taken place as people with cognitive disabilities moved from institutions to public schools and community living settings [3]. In recognition of these changes, the Americans with Disabilities Act (ADA) was enacted into law, to encourage integration and eliminate discrimination against individuals with disabilities in critical areas including employment, housing, transportation, recreation, health services, and access to public services [4]. Enforcement of this federal accessibility legislation requires public transportation operators to make their vehicles accessible to people with disabilities [5], better quality, and improve assistive device technologies.

Researchers and App developers have tried to address the problem of improving information access to travelers. Also, commercial firms like [6], specialize in retrofitting existing bus fleets with such technologies to enhance fleet management and accountability. Transportation apps were first developed for the traveling public in urban areas, giving transit customers access to bus schedules and real-time bus arrival information. The development of these apps was an initiative by Google to incorporate fixed-route transit data into its Google Maps program. In 2005, the General Transit Feed Specification (GTFS) was introduced. It was a result of a project between Google and TriMet in Portland to create a transit trip-planner using the Google Maps web application [7]. The specification was designed to be simple for agencies to create, easy for programmers to access, and comprehensive enough to describe an intricate transit system [8]. GTFS identifies a series of comma-separated files which together describe the stops, trips, routes and fare information about an agency's service. Once transit agencies had geo-coded their routes and schedules for Google Maps, taking the next step of using these data in real-time transit technologies for mobile devices, such as smartphones, was easy. Apps developed by using these technologies allowed the creation of data streams that revealed the exact location at any given time of the bus/train and the passenger as well as lines not operating or experiencing delays. Thus, an app could locate a user's current position, find the closest local fixed-route transit stop, and show the user in real-time when the next bus or train would arrive at that stop. This type of data allowed for the creation of maps, routes, schedules, and interactive sites for trip planning, and, eventually, specific transit apps.

Transit or mobility Apps help to assist users in planning or understanding their transportation choices and may enhance access to alternative modes. In many ways, transit apps have swiftly moved through their evolution to meet ever more demanding users and to take advantage of more sophisticated data streams and phones. Today, transportation apps can suggest alternative routes and modes based on real-time traffic and service information; and enable users to pay their fares with a tap or swipe.

According to [9], transit apps can ease some cognitive difficulties, primarily by reducing the cognitive effort required to make sense of complex situations and the steps required to achieve specific tasks. Several studies have shown that bus riders without real-time arrival data perceived wait times to be longer than was felt by riders with real-time data, suggesting that real-time information can increase the perceived trip satisfaction [10] Therefore, this helps both the average user and those with cognitive challenges and disabilities. Behavioral mechanisms from the disciplines of psychology and economics are being used in mobile apps to help users. Apps deliver their cognitive benefits both through the physical nature of any mediated service and through the design of the app itself. Transportation apps often involve search and decision heuristics, and how these are designed can have important impacts on whether or not the app is used and therefore whether or not it changes behavior. A well-designed app will improve search heuristics by reducing the cognitive load necessary to make efficient and effective searches. It may also improve decision heuristics by making it easy to sort through options and come to a decision. Additionally, with an aging population in the developed world, heuristic practices for older adults that account for declining vision, hearing, motor skills, and cognitive function will become of increasing importance to ensure that transportation apps meet the mobility needs of disabled and older populations [11] [12]. Table 1. below provides an overview of these mechanisms and the types of apps currently employing them [13].

Table 1 Benefits of Mobility Apps (FHWA, 2016)

Behavioral Mechanism and Benefit	**Mobility App Example**
Alleviating cognitive burdens with powerful search tools	Google Maps, Apple Maps
Improving actual and perceived traveler control over journeys	OneBusAway
Improving trust in carpooling services	Carma
Changing norms around transportation, such as the ease of mobile ticketing	GlobeSherpa
Impacting price directly by enabling competitive services	Uber, Lyft
Changing perceptions of value across multiple modes	RideScout
Improving information availability and shaping service usage	Transit App, Transloc
Harnessing existing social pressures and generating new ones to shape travel behavior in the desired direction	Waze
Delivering Financial and non-financial incentives in favor of one behavior or another	GasBuddy

Apps like CityMapper, Moovit, OneBusAway, Transit, Transloc, and many others have made it easier for transit users but fails to address the challenges of the Vulnerable Transit Users like the elderly and disabled persons. CityMapper app provides support for taxis, subways, trains, ferries, cabs, and even Uber and Lyft. It also helps one get departure times, alerts, step-by-step directions, and offline maps in some areas. OneBusAway helps people find bus stops in an area but is limited to a few cities. It sources arrival times with the local transportation agency. Moovit provides train schedules, bus schedules, subway schedules, and trams but does not provide rides. Transit provides step-by-step navigation, bike-sharing stations, schedules for buses, subways and integration with Uber

2 The Proposed Model/Implementation

The app was developed for the Android smartphone platform using Android Studio, Java, node.js, and Firebase's Cloud Firestore for database management and crowd-sourcing capabilities. The Firebase Realtime Database sources the dynamic data from the backend and moves markers on the map in response to real-time transitions. Map data was sourced from Google Maps API, with routing being implemented using custom code.

In this proposed model, Android has been chosen for the application because it is applicable for multiple devices and also with a wide range of prices for different models and specifications of the phone. This mobile application tracks moving buses and visualizes them on a map.

Figure 1. Below provides an overview of the system's linkage of different components in terms of software and Technology

Figure 1 The Proposed Architecture

The proposed system uses the General Transit Feed Specification (GTFS) or static transit and GTFS Real-time as the standard of defining the data's format to easily communicate with the bus's agency system. The GTFS file provided by the bus agencies consists of information such as bus schedules, bus trips, bus stops, bus routes, fares, and other information. GTFS Real-time is a feed specification that allows public transportation agencies to provide real-time updates about their fleet to application developers. GTFS Real-time supports trip updates, service alerts, and vehicle positions. The GTFS files are stored into an in-memory SQLite database, so that we can then run SQL queries over the data to figure out where each bus should be at the given times and later be published into a Firebase Real-time Database. This is necessary to optimize the data retrieval from the web and mobile app with the execution of SQL statements through the Google Maps Directions API. It allows the optimization of network resources with just only requests for the data from the database required by the users at that moment.

The Firebase Realtime Database is cloud-hosted database. The data is stored in a JavaScript Object Notation (JSON) format and synchronized in real-time to every connected client. A Firebase Realtime Database stores the vehicle locations, and vehicles are snapped to the road with the Roads API. Firebase provides real-time data synchronization to the backend and map. The App uses a Firebase Real-time Database to communicate location updates between the various components of the server and front-end applications. When the vehicle locator, or the simulator, stores the location updates in the Firebase Real-time Database, Firebase sends automatic updates to the backend, which in turn updates the front-end display. Figure 2 below shows the real-time firebase database interface with data information.

Figure 2 Interface of the Real-time Firebase Database

3.1 Backend

The backend is a code that runs on a server when requests are received from the clients. It contains the logic to send the appropriate data back to the client. The app backend server was built using the Google cloud platform. The real-time firebase which stores and sorts the important information that the end-user does not see is also part of the backend. The backend was set up by signing into Google Cloud Platform console (console.cloud.google.com) and creating a new project. After creating the Project name, a Project ID was then allocated which is a unique name across all Google Cloud projects. The Google cloud platform has a command line environment running in the Cloud called Cloud Shell. Cloud Shell is a Linux virtual machine that's loaded with all the development tools. The backend is built with Node JavaScript (Node.js). Node.js is important for building real-time applications. The backend processes locations from the Firebase Real-time Database and predicts travel times using the Google Directions API. A service call, like selecting status online, fetches data from the backend. This places the data on an

immutable model stream. An Interactor listening to this stream notices the new data and passes it to the Presenter. The Presenter formats the data and sends it to the View.

3.1 Frontend

Android Studio was used to develop the frontend of the mobile application. Android studio is the official integrated development environment for Google's Android operating system and open-source software or tool for the development of android applications. It is an integrated development environment (IDE), which has a strong code editing tool for developing a creative user interface (UI) and user controls (UX), and an Android Virtual Device (Emulator) to run and debug apps without the need for a physical device.

Figure 3. Below shows an android studio interface with custom code showing the frontend design

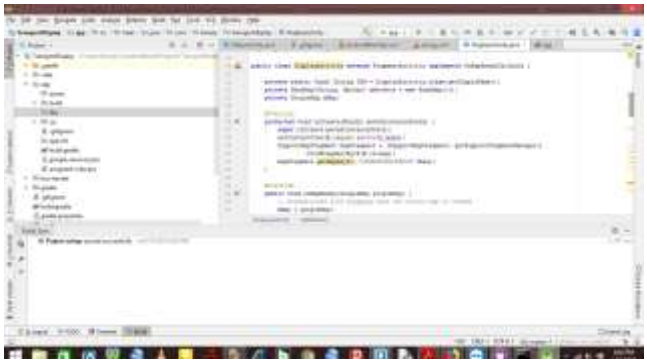


Figure 3 Interface of Android Studio

4 Results

4.1 Driver App

The driver app was developed based on the logic shown in Figure 5 below. When a driver selects the status to be online, regardless of whether the application has an activity, the Google map navigates the driver to where services have been requested. That is, the location of the driver is sent to the user/rider for real-time tracking. The app makes it easy to find your customers/ riders and navigate to their destination. Running the app on android smartphone requests a driver to log in and indicate whether online or offline by hitting the on.

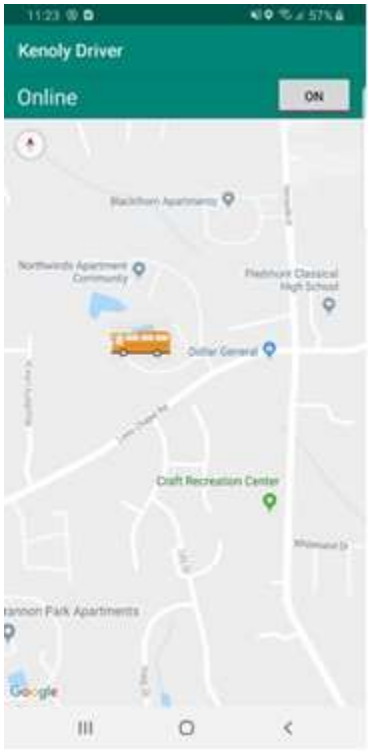


Figure 4 Interface design of the Driver App

4.2 User App

The user app provides these paratransit subscribers a dial-up button for making bus appointments, a real-time bus schedule information for other transit agencies and a map for visualizing and tracking requested buses. It enables users to track the driver's location and which way he is traveling. In case, the driver has not moved from a location or, they are taking a long route, the users can always call them and direct them to the pick-up location. When the correct bus approaches the pickup point, within approximately 20 ft. from the user, the system alerts the user via synthetic speech that the designated bus is arriving.

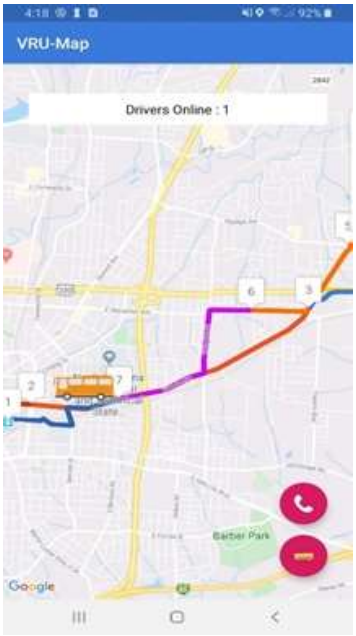


Figure 5 Interface design of Rider App

4 Conclusion/Future Works

In this paper, the challenges facing vulnerable road users in patronizing paratransit services are addressed. It provides a customizable tracking map, accessible information to users of paratransit transportation who are usually mobility impaired or have cognitive impairments. Future research will include; evaluation of the apps, presence of different mode type selection (multimodal), augmented reality map for walk navigation to bus stops, in app-chat option, integrated payment options, multi-language support, preferred driver selection option, real-time weather information and customer service support feedback.

ACKNOWLEDGMENTS

The authors are thankful to the Center for Advance Transportation Mobility and the North Carolina A&T State University for providing funds to pursue this research.

REFERENCES

[1] FHWA- Federal Highway Administration. 2016. *Smartphone Applications to Influence Travel Choices: Practices and Policies*. Washington D.C: U.S. Department of Transportation.

[2] ADA. 1990. "*ADA* (Americans with Disabilities Act) *Network*. Retrieved from https://adata.org/faq/what-definition-disability-under-ada

[3] Braddock, D. 2002. Public Financial Support for Disability at the Dawn of the 21st Century. In D. Braddock (ed.): *Disability at the Dawn of the 21st Century and the State of the States* . Washington, DC: American Association on Mental Retardation, pp. 65-76..

[4] Braddock, D., Rizzolo, M. C., Thompson, M., AND Bell, R. 2005. "Emerging technologies and cognitive disability." *J. Spec. Educ. Tech.*

[5] Calak, P. 2013. "Smartphone Evaluation Heuristics for Older Adults."

[6] Google Inc. 2012. *General Transit Feed Specification Reference.* http://developers.google.com/transit/gtfs/reference.

[7] Intuicom. 2003. *Intelligent Transportation.* https://www.intuicom.com/.

[8] Marczewski, Andrzej. 2012. *Gamification: A Simple Introduction and a Bit More*. Seattle, WA: Amazon Digital Services..

[9] Roth. 2010. ""How Google and Portland's TriMet Set the Standard for Open."

[10] Silva P.A., Holden K., Nii A. 2014. Smartphones, Smart Seniors, But Not-So-Smart Apps: *A Heuristic Evaluation of Fitness Apps*. In: Schmorrow D.D., Fidopiastis C.M. (eds) Foundations of Augmented Cognition. Advancing Human Performance and Decision-Making through Adaptive Systems. AC 2014. Lecture Notes in Computer Science, vol 8534. Springer, Cham. https://doi.org/10.1007/978-3-319-07527-3_33.

[11] Szczerba, R. 2014. "Using Technology To Mitigate Cognitive Disabilities." *Forbes.* https://www.forbes.com/sites/robertszczerba/2014/07/30/using-technology-to-mitigate-cognitive-disabilities/#15ba53877619.

[12] U.S. Department of Transportation 2007. *Nondiscrimination on the Basis of Disability in Air Travel; Final Rule*..Retrieved from https://www.govinfo.gov/content/pkg/FR-2008-05-13/pdf/08-1228.pdf#page=53

[13] United States Government Accountability Office. 2012. "ADA Pa2ratransit Services: Demand Has Increased, but Little is Known about Compliance." Washington D.C.

Author List

Author List(Continued)

The 24th International Database
&
Applications Engineering Symposium

ISBN 978-1-4503-7503-0